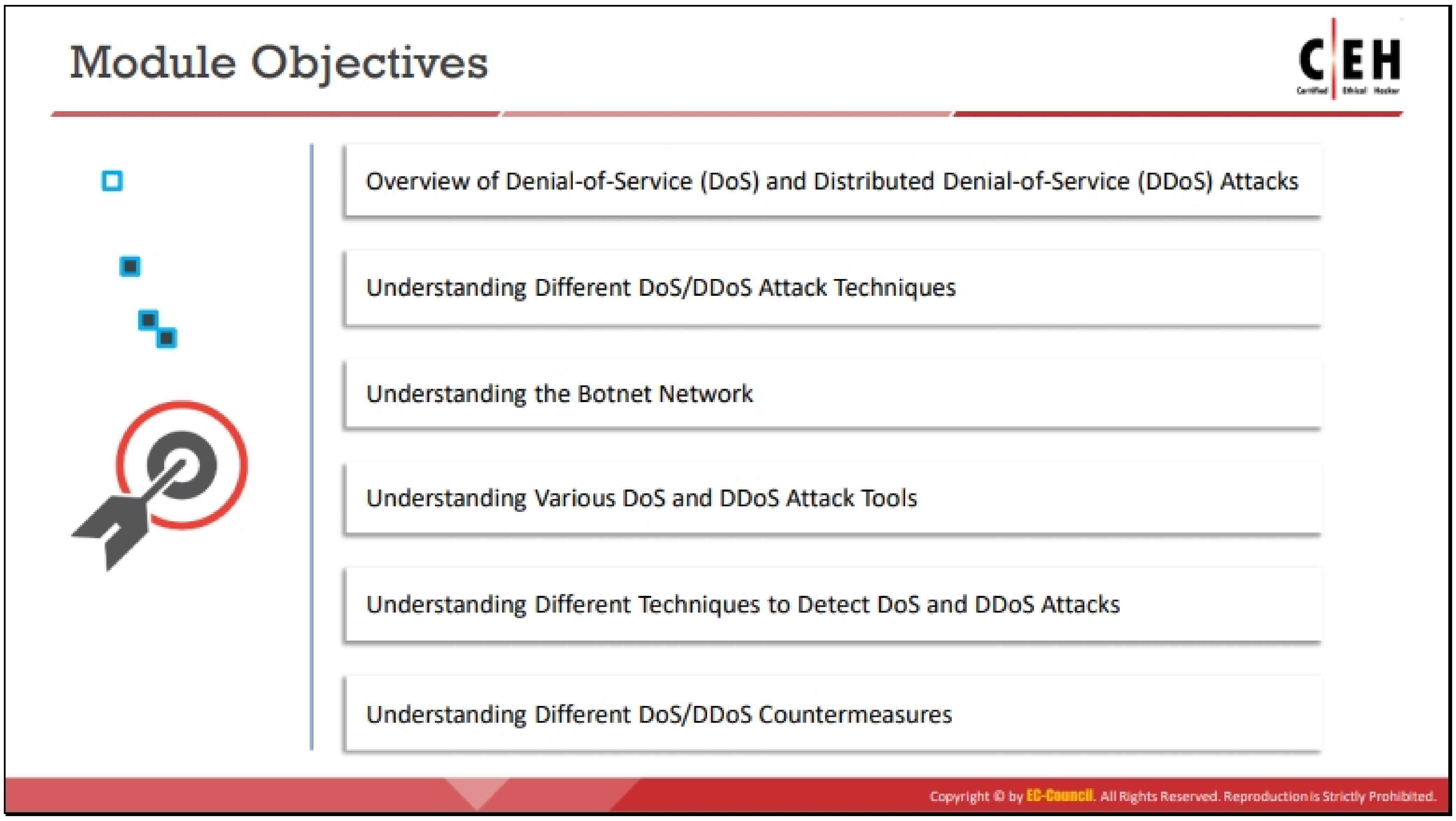
C|EH™
Certified Ethical Hacker

Module 10:
Denial-of-Service



Module Objectives

- Overview of Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) Attacks
- Understanding Different DoS/DDoS Attack Techniques
- Understanding the Botnet Network
- Understanding Various DoS and DDoS Attack Tools
- Understanding Different Techniques to Detect DoS and DDoS Attacks
- Understanding Different DoS/DDoS Countermeasures

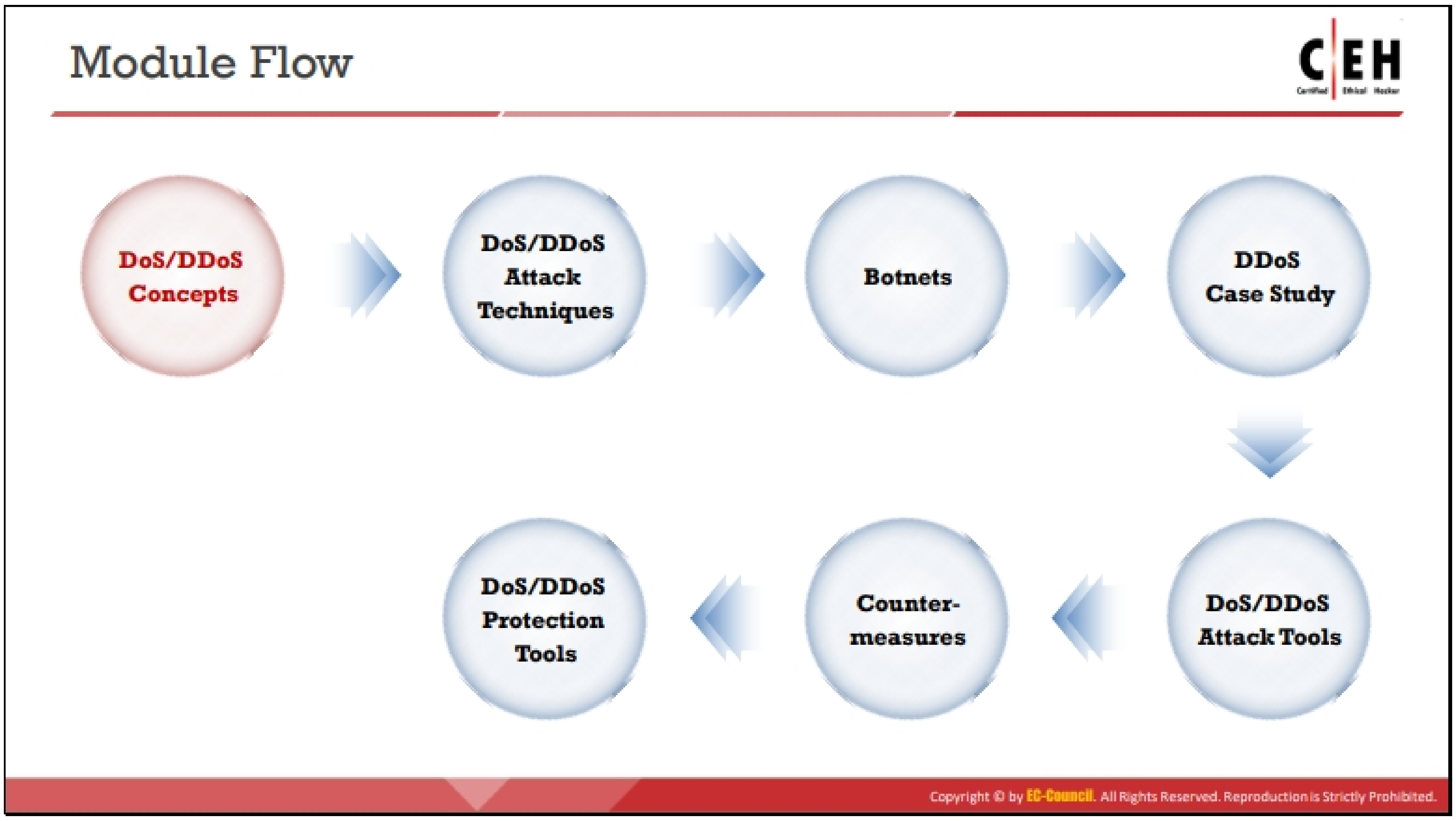
Module Objectives

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks are a major threat to computer networks. These attacks attempt to make a machine or network resource unavailable to its authorized users. Usually, DoS/DDoS attacks exploit vulnerabilities in the implementation of the Transmission Control Protocol (TCP)/Internet Protocol (IP) model or bugs in a specific operating system (OS).

This module starts with an overview of DoS and DDoS attacks and then provides insight into different DoS/DDoS attack techniques. Later, it discusses the botnet network, DoS/DDoS attack tools, techniques to detect DoS/DDoS attacks, and DoS/DDoS countermeasures.

At the end of this module, you will be able to do the following:

- Describe DoS/DDoS concepts
- Understand various DoS/DDoS attack techniques
- Describe botnets
- Illustrate DoS/DDoS case studies
- Explain different DoS/DDoS attack tools
- Apply best practices to mitigate DoS/DDoS attacks



Module Flow

DoS/DDoS Concepts

DoS/DDoS Attack Techniques

Botnets

DDoS Case Study

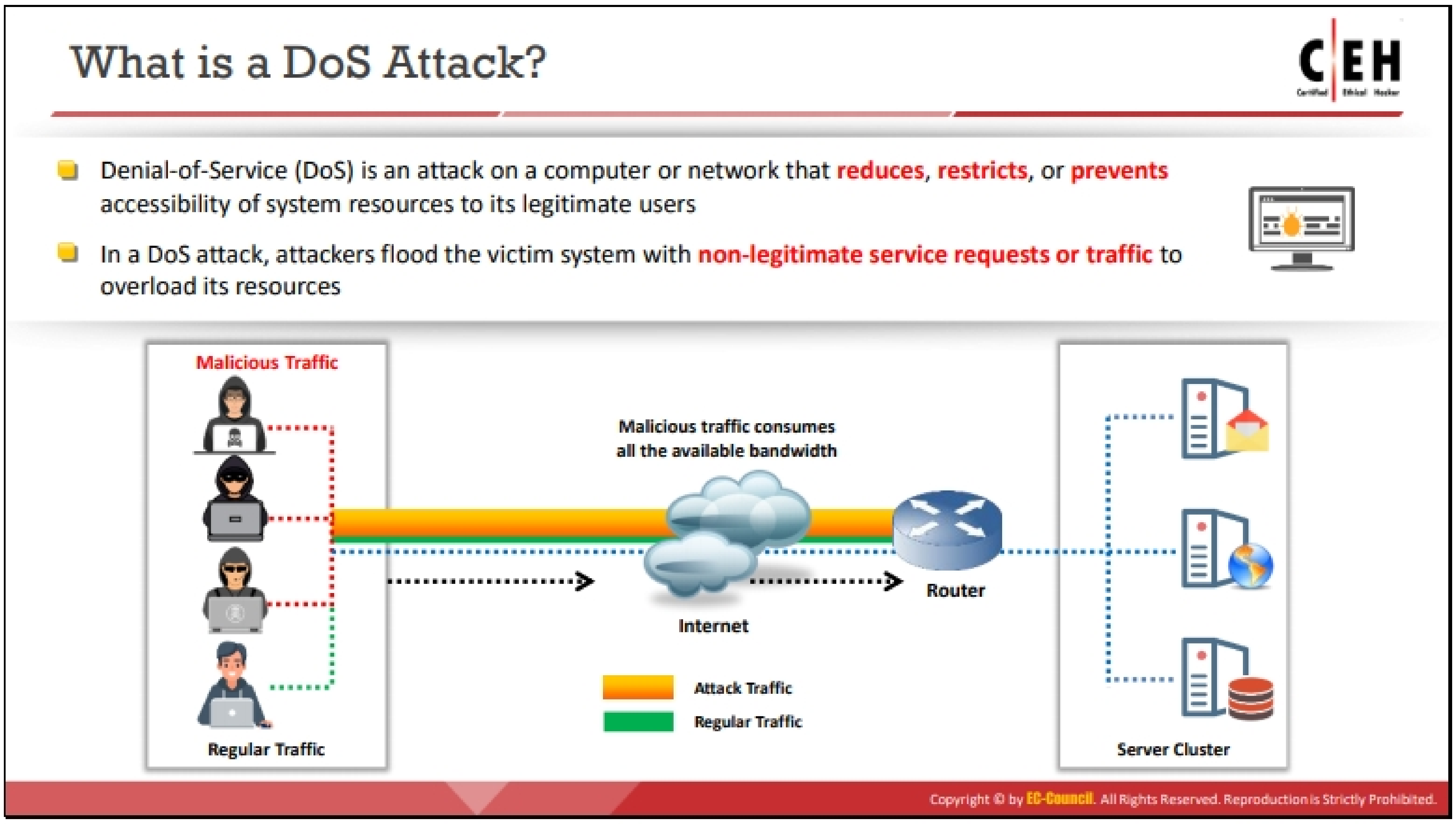
DoS/DDoS Attack Tools

Counter-measures

DoS/DDoS Protection Tools

DoS/DDoS Concepts

For a good understanding of DoS/DDoS attacks, one must be familiar with related concepts in advance. This section defines DoS and DDoS attacks and discusses how DDoS attacks work.



What is a DoS Attack?

- Denial-of-Service (DoS) is an attack on a computer or network that **reduces**, **restricts**, or **prevents** accessibility of system resources to its legitimate users
- In a DoS attack, attackers flood the victim system with **non-legitimate service requests or traffic** to overload its resources

🟧 Attack Traffic
🟩 Regular Traffic

What is a DoS Attack?

A DoS attack is an attack on a computer or network that reduces, restricts, or prevents access to system resources for legitimate users. In a DoS attack, attackers flood a victim's system with nonlegitimate service requests or traffic to overload its resources and bring down the system, leading to the unavailability of the victim's website or at least significantly reducing the victim's system or network performance. The goal of a DoS attack is to keep legitimate users from using the system, rather than to gain unauthorized access to a system or to corrupt data.

The following are examples for types of DoS attacks:

- Flooding the victim's system with more traffic than it can handle
- Flooding a service (e.g., Internet Relay Chat (IRC)) with more events than it can handle
- Crashing a TCP/IP stack by sending corrupt packets
- Crashing a service by interacting with it in an unexpected manner
- Hanging a system by causing it to go into an infinite loop

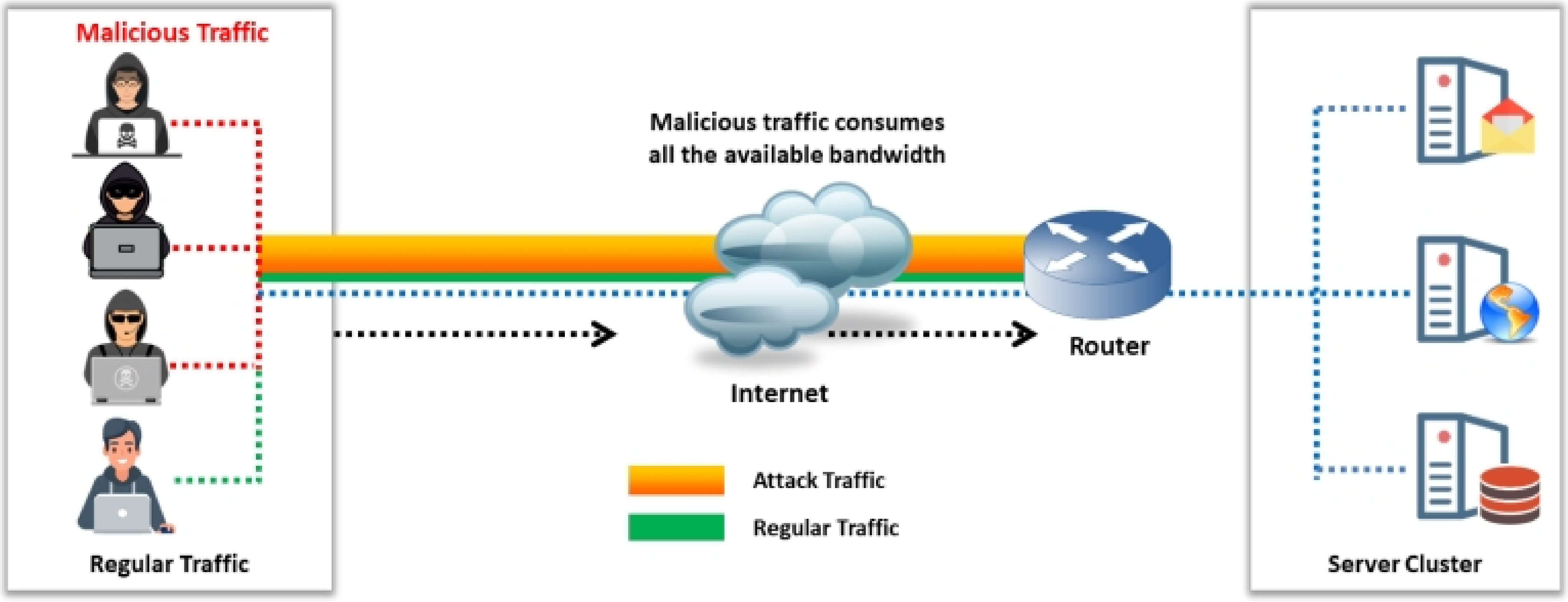


Figure 10.1: Schematic of a DoS attack

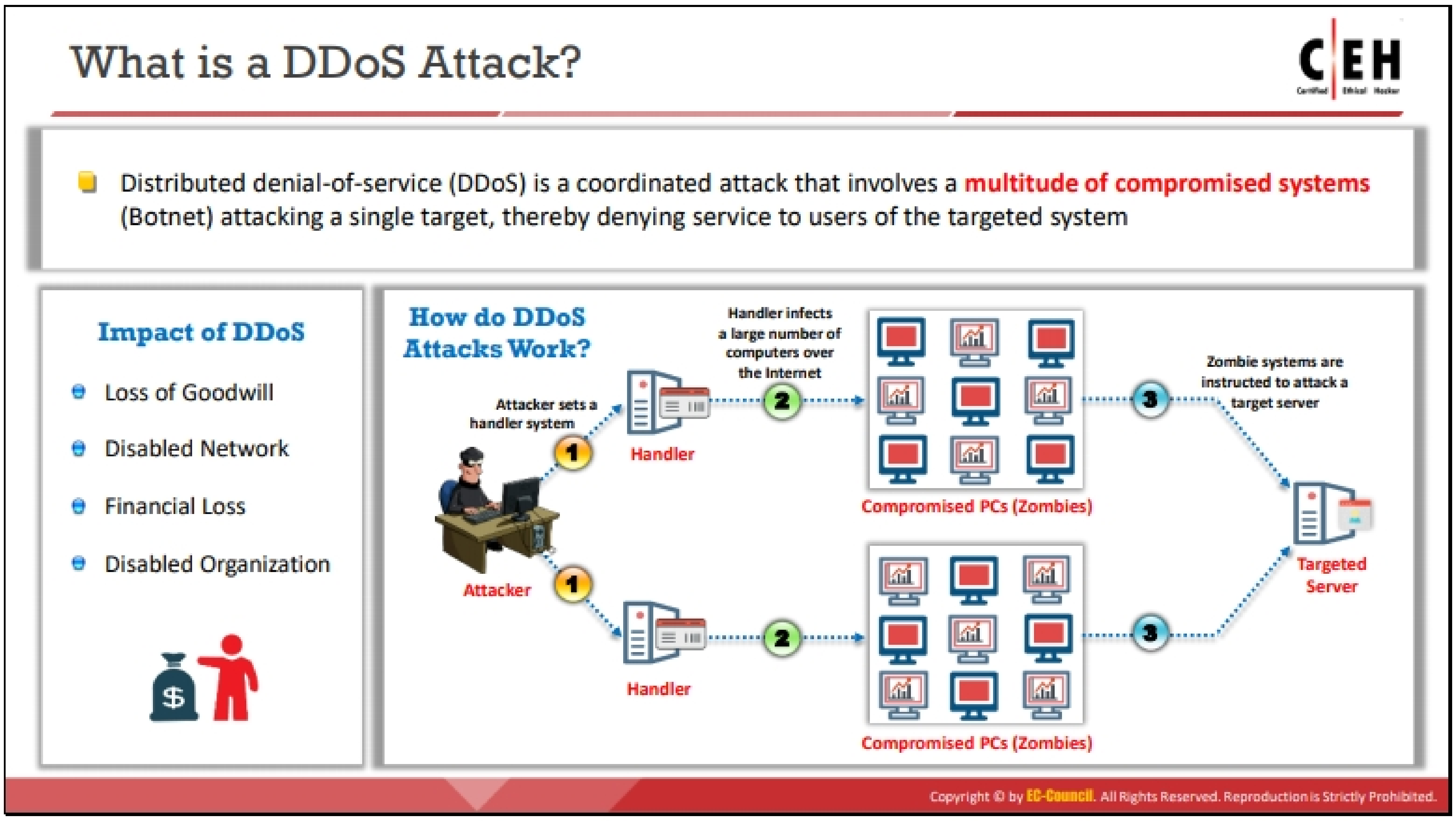
DoS attacks have various forms and target various services. The attacks may cause the following:

- Consumption of resources
- Consumption of bandwidth, disk space, CPU time, or data structures
- Actual physical destruction or alteration of network components
- Destruction of programming and files in a computer system

In general, DoS attacks target network bandwidth or connectivity. Bandwidth attacks overflow the network with a high volume of traffic by using existing network resources, thereby depriving legitimate users of these resources. Connectivity attacks overflow a system with a large number of connection requests, consuming all available OS resources to prevent the system from processing legitimate user requests.

Consider a food catering company that conducts much of its business over the phone. If an attacker wants to disrupt this business, they need to find a way to block the company's phone lines, which would make it impossible for the company to do business. A DoS attack works along the same lines—the attacker uses up all the ways to connect to the victim's system, making legitimate business impossible.

DoS attacks are a kind of security breach that does not generally result in the theft of information. However, these attacks can harm the target in terms of time and resources. Furthermore, security failure might cause the loss of a service such as email. In the worst-case scenario, a DoS attack can cause the accidental destruction of the files and programs of millions of people who were connected to the victim's system at the time of the attack.



What is a DDoS Attack?

- Distributed denial-of-service (DDoS) is a coordinated attack that involves a **multitude of compromised systems** (Botnet) attacking a single target, thereby denying service to users of the targeted system

Impact of DDoS

- Loss of Goodwill
- Disabled Network
- Financial Loss
- Disabled Organization

How do DDoS Attacks Work?

Attacker sets a handler system
Handler infects a large number of computers over the Internet
Zombie systems are instructed to attack a target server
1
2
3
Attacker
Handler
Handler
Compromised PCs (Zombies)
Compromised PCs (Zombies)
Targeted Server
$

What is a DDoS Attack?

Source: *http://searchsecurity.techtarget.com*

A DDoS attack is a large-scale, coordinated attack on the availability of services on a victim's system or network resources, and it is launched indirectly through many compromised computers (botnets) on the Internet.

As defined by the World Wide Web Security FAQ, "A distributed denial-of-service (DDoS) attack uses many computers to launch a coordinated DoS attack against one or more targets. Using client/server technology, the perpetrator is able to multiply the effectiveness of the denial of service significantly by harnessing the resources of multiple unwitting accomplice computers, which serve as attack platforms." The flood of incoming messages to the target system essentially forces it to shut down, thereby denying service to legitimate users.

The services under attack belong to the "primary victim," whereas the compromised systems used to launch the attack are called "secondary victims." The use of secondary victims in performing a DDoS attack enables the attacker to mount a large and disruptive attack while making it difficult to track down the original attacker.

The primary objective of a DDoS attack is to first gain administrative access on as many systems as possible. In general, attackers use a customized attack script to identify potentially vulnerable systems. After gaining access to the target systems, the attacker uploads and runs DDoS software on these systems at the time chosen to launch the attack.

DDoS attacks have become popular because of the easy accessibility of exploit plans and the negligible amount of brainwork required to execute them. These attacks can be very dangerous because they can quickly consume the largest hosts on the Internet, rendering them useless.

The impacts of DDoS include the loss of goodwill, disabled networks, financial losses, and disabled organizations.

How do DDoS Attacks Work?

In a DDoS attack, many applications barrage a target browser or network with fake exterior requests that make the system, network, browser, or site slow, useless, and disabled or unavailable.

The attacker initiates the DDoS attack by sending a command to zombie agents, which are Internet-connected computers compromised by an attacker through malware programs to perform various malicious activities through a command and control (C&C) server. These zombie agents send a connection request to a large number of reflector systems with the spoofed IP address of the victim, which causes the reflector systems to presume that these requests originate from the victim's machine instead of the zombie agents. Hence, the reflector systems send the requested information (response to the connection request) to the victim. Consequently, the victim's machine is flooded with unsolicited responses from several reflector computers simultaneously, which may either reduce the performance or cause the victim's machine to shut down completely.

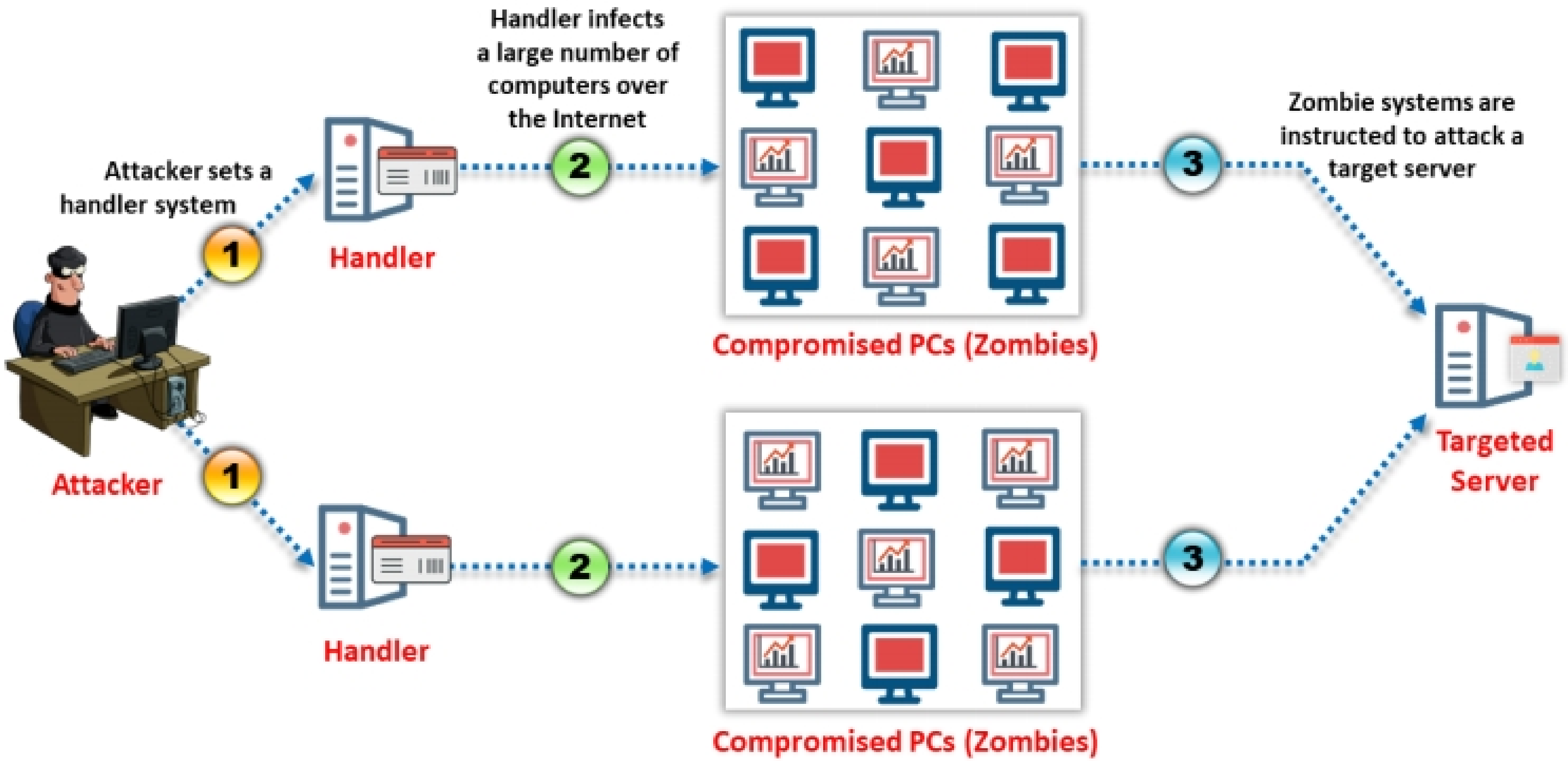
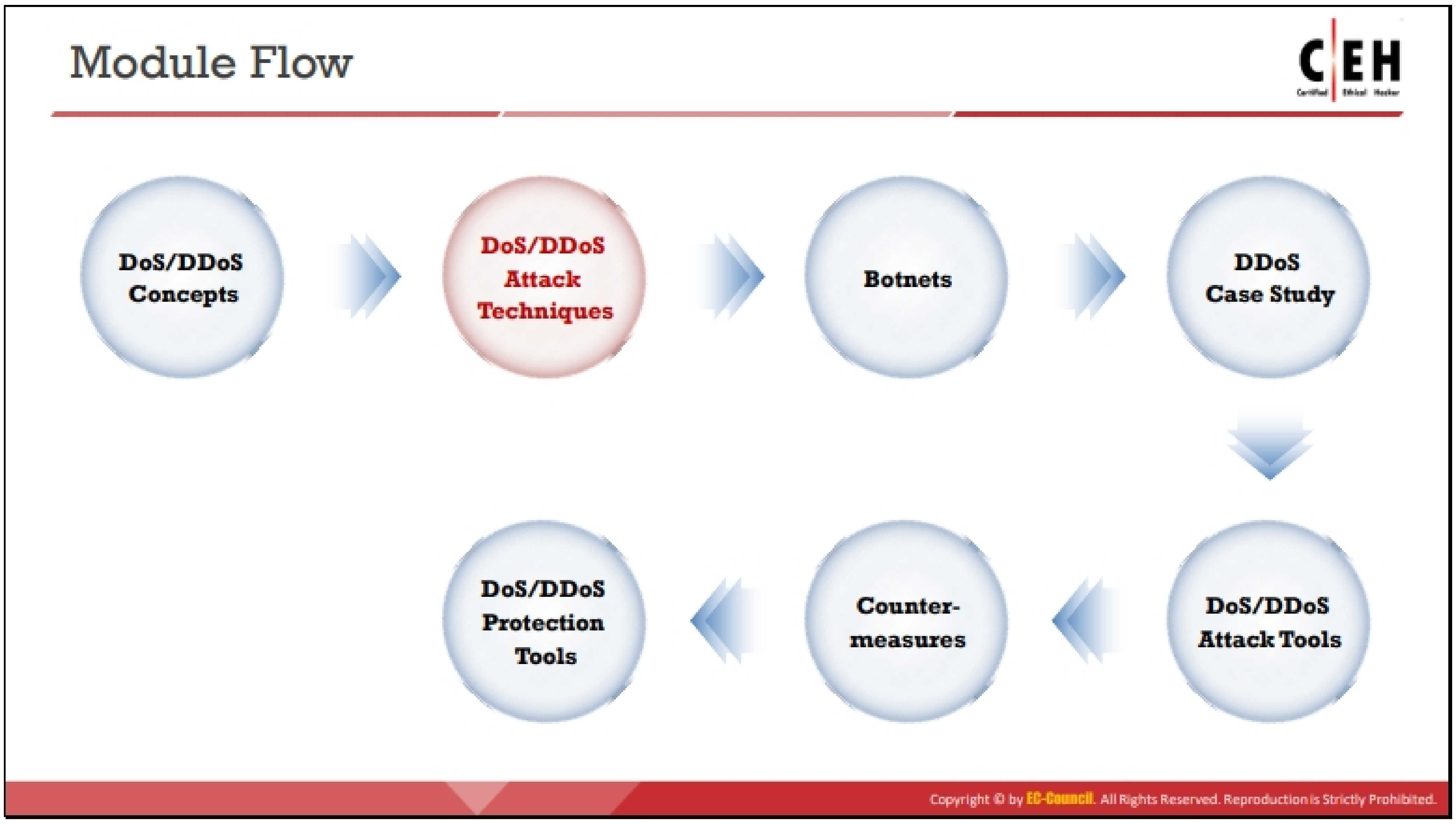

Handler infects a large number of computers over the Internet
Attacker sets a handler system
Zombie systems are instructed to attack a target server
1
2
3
Handler
Compromised PCs (Zombies)
Attacker
Targeted Server
1
2
3
Handler
Compromised PCs (Zombies)

Figure 10.2: Schematic of a DDoS attack

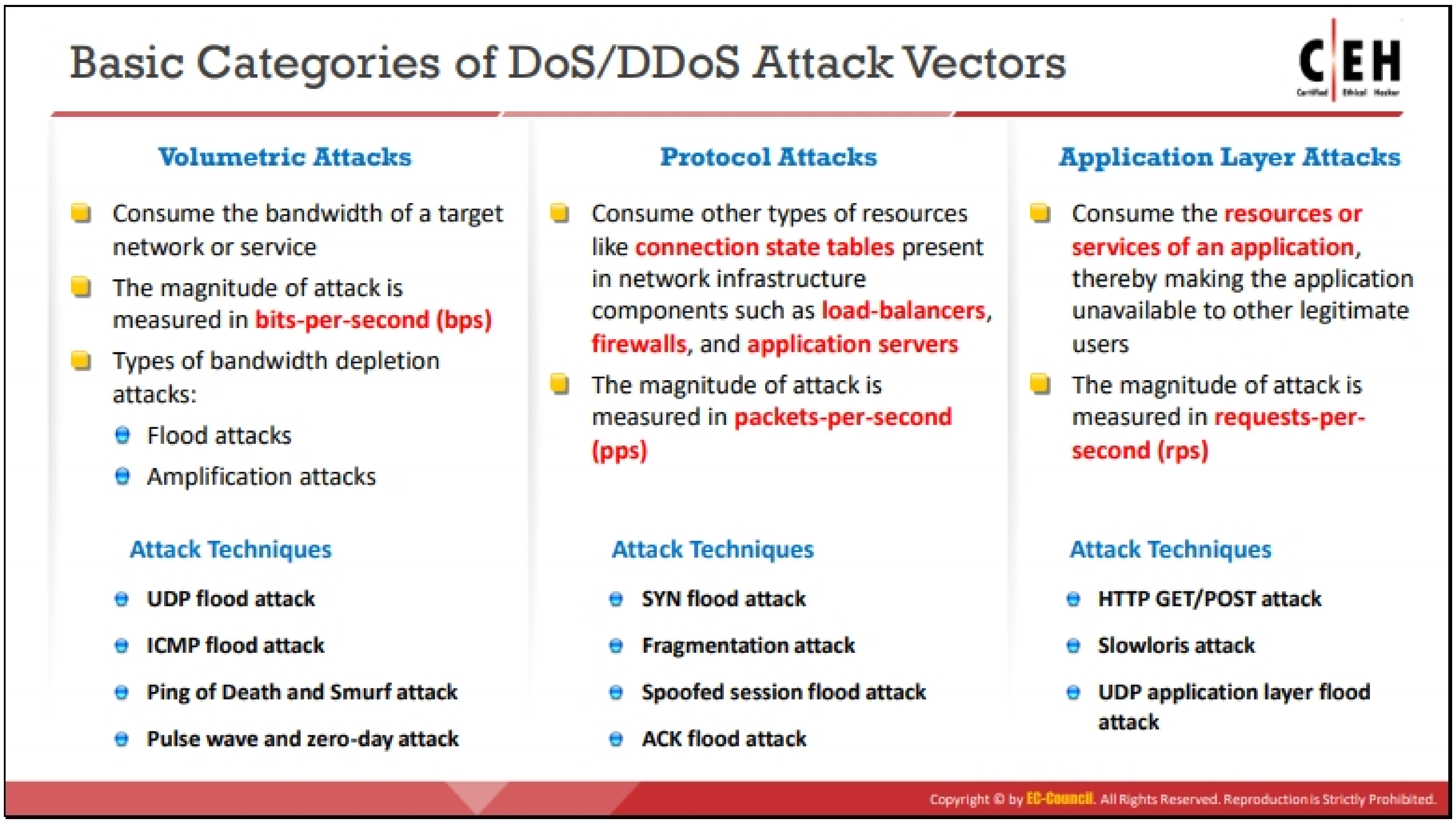


Module Flow

DoS/DDoS Concepts
DoS/DDoS Attack Techniques
Botnets
DDoS Case Study
DoS/DDoS Attack Tools
Counter-measures
DoS/DDoS Protection Tools
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

DoS/DDoS Attack Techniques

Attackers implement various techniques to launch DoS/DDoS attacks on target computers or networks. This section discusses the basic categories of DoS/DDoS attack vectors and various attack techniques.



Basic Categories of DoS/DDoS Attack Vectors

Volumetric Attacks

- Consume the bandwidth of a target network or service
- The magnitude of attack is measured in **bits-per-second (bps)**
- Types of bandwidth depletion attacks:
  - Flood attacks
  - Amplification attacks

Attack Techniques

- **UDP flood attack**
- **ICMP flood attack**
- **Ping of Death and Smurf attack**
- **Pulse wave and zero-day attack**

Protocol Attacks

- Consume other types of resources like **connection state tables** present in network infrastructure components such as **load-balancers**, **firewalls**, and **application servers**
- The magnitude of attack is measured in **packets-per-second (pps)**

Attack Techniques

- **SYN flood attack**
- **Fragmentation attack**
- **Spoofed session flood attack**
- **ACK flood attack**

Application Layer Attacks

- Consume the **resources or services of an application**, thereby making the application unavailable to other legitimate users
- The magnitude of attack is measured in **requests-per-second (rps)**

Attack Techniques

- **HTTP GET/POST attack**
- **Slowloris attack**
- **UDP application layer flood attack**

Basic Categories of DoS/DDoS Attack Vectors

DDoS attacks mainly aim to diminish the network bandwidth by exhausting network, application, or service resources, thereby restricting legitimate users from accessing system or network resources. In general, DoS/DDoS attack vectors are categorized as follows:

- **Volumetric Attacks**

  These attacks exhaust the bandwidth either within the target network/service or between the target network/service and the rest of the Internet to cause traffic blockage, preventing access to legitimate users. The attack magnitude is measured in bits per second (bps).

  Volumetric DDoS attacks generally target protocols such as the Network Time Protocol (NTP), Domain Name System (DNS), and Simple Service Discovery Protocol (SSDP), which are stateless and do not have built-in congestion avoidance features. The generation of a large number of packets can cause the consumption of the entire bandwidth on the network. A single machine cannot make enough requests to overwhelm network equipment. Hence, in DDoS attacks, the attacker uses several computers to flood a victim. In this case, the attacker can control all the machines and instruct them to direct traffic to the target system. DDoS attacks flood a network, causing a significant statistical change in network traffic that overwhelms network equipment such as switches and routers. Attackers use the processing power of a large number of geographically distributed machines to generate huge traffic directed at the victim, which is why such an attack is called a DDoS attack.

There are two types of bandwidth depletion attacks:

- In a **flood attack**, zombies send large volumes of traffic to the victim's systems to exhaust the bandwidth of these systems.
- In an **amplification attack**, the attacker or zombies transfer messages to a broadcast IP address. This method amplifies malicious traffic that consumes the bandwidth of the victim's systems.

Attackers use botnets and perform DDoS attacks by flooding the network. The entire bandwidth is used up by attackers, and no bandwidth remains for legitimate use. The following are examples for volumetric attack techniques:

- User Datagram Protocol (UDP) flood attack
- Internet Control Message Protocol (ICMP) flood attack
- Ping of Death (PoD) attack
- Smurf attack
- Pulse wave attack
- Zero-day attack
- Malformed IP packet flood attack
- Spoofed IP packet flood attack

- **Protocol Attacks**

Attackers can also prevent access to a target by consuming types of resources other than bandwidth, such as connection state tables. Protocol DDoS attacks exhaust resources available on the target or on a specific device between the target and the Internet. These attacks consume the connection state tables present in network infrastructure devices such as load balancers, firewalls, and application servers. Consequently, no new connections will be allowed, because the device will be waiting for existing connections to close or expire. In this case, the attack magnitude is measured in packets per second (pps) or connections per second (cps). These attacks can even take over the state of millions of connections maintained by high-capacity devices.

The following are examples for protocol attack techniques:

- Synchronize (SYN) flood attack
- Fragmentation attack
- Spoofed session flood attack
- Acknowledgement (ACK) flood attack
- SYN-ACK flood attack
- ACK and PUSH ACK flood attack
- TCP connection flood attack
- TCP state exhaustion attack
- RST attack

- **Application Layer Attacks**

  In these attacks, the attacker attempts to exploit vulnerabilities in the application layer protocol or in the application itself to prevent legitimate users from accessing the application. Attacks on unpatched, vulnerable systems do not require as much bandwidth as protocol or volumetric DDoS attacks for succeeding. In application DDoS attacks, the application layer or application resources are consumed by opening connections and leaving them open until no new connections can be made. These attacks destroy a specific aspect of an application or service and can be effective with one or a few attacking machines that produce a low traffic rate. Furthermore, these attacks are very difficult to detect and mitigate. The magnitude of attack is measured in requests per second (rps).

  Application-level flood attacks result in the loss of services of a particular network, such as emails and network resources, or the temporary shutdown of applications and services. Through this attack, attackers exploit weaknesses in programming source code to prevent the application from processing legitimate requests.

  Several kinds of DoS attacks rely on software-related exploits such as buffer overflows. A buffer overflow attack sends excessive data to an application that either shuts down the application or forces the data sent to the application to run on the host system. The attack crashes a vulnerable system remotely by sending excessive traffic to an application.

  Occasionally, attackers can also execute arbitrary code on the remote system via a buffer overflow. Sending too much data to an application overwrites the data that controls the program, enabling the hacker to run their code instead.

  Using application-level flood attacks, attackers attempt to do the following:

  - Flood web applications with legitimate user traffic
  - Disrupt service to a specific system or person by, for example, blocking a user's access through repeated invalid login attempts
  - Jam the application database connection by crafting malicious Structured Query Language (SQL) queries

  Application-level flood attacks can result in a substantial loss of money, service, and reputation for organizations. These attacks occur after the establishment of a connection. Because a connection is established and the traffic entering the target appears to be legitimate, it is difficult to detect these attacks. However, if the user identifies the attack, they can stop it and trace it back to its source more easily than other types of DDoS attacks.

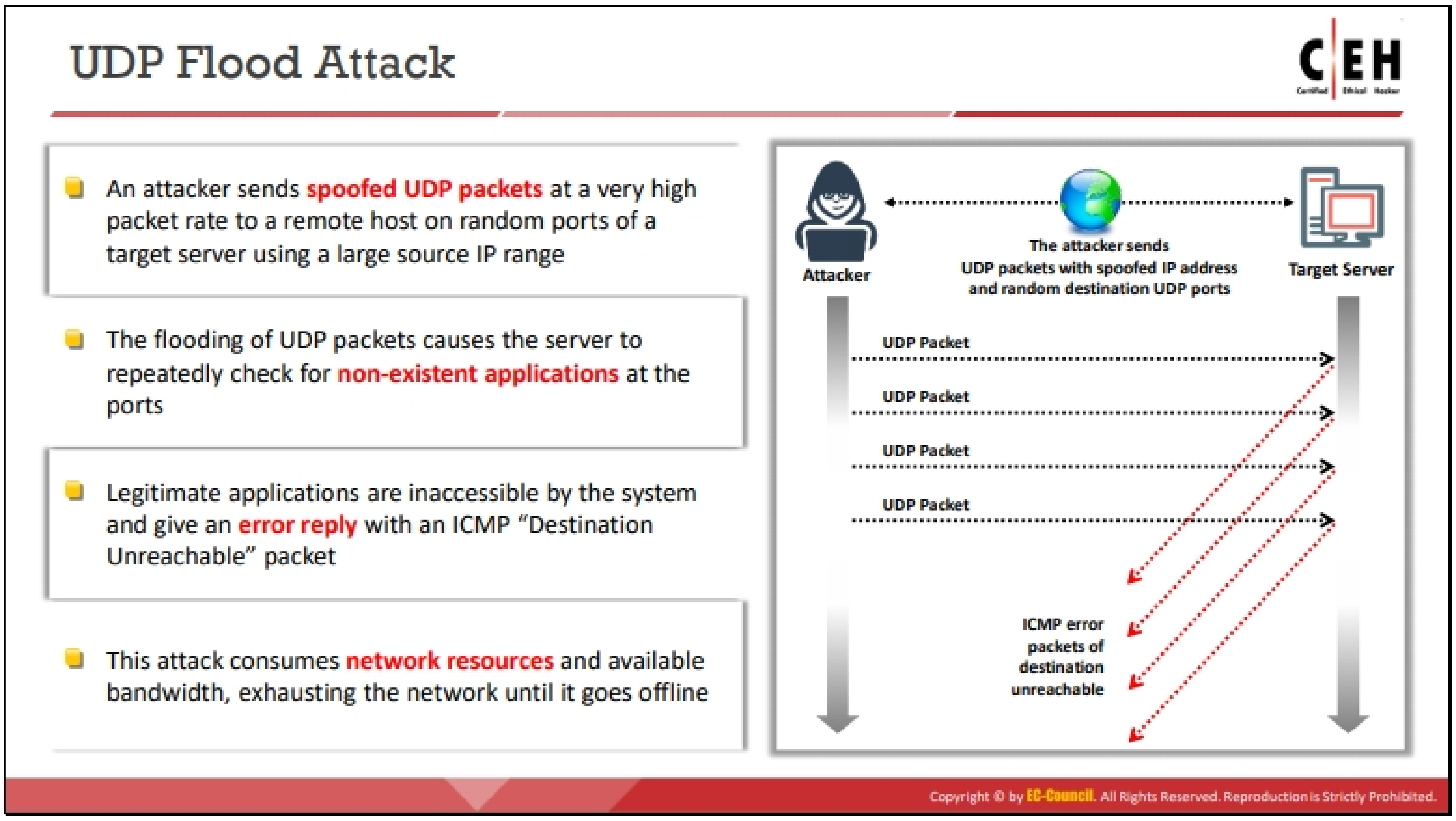
  The following are examples for application layer attack techniques:

  - Hypertext Transfer Protocol (HTTP) flood attack
  - Slowloris attack
  - UDP application layer flood attack

DoS/DDoS Attack Techniques

Next, the following DoS/DDoS attack techniques will be discussed:

- UDP flood attack
- ICMP flood attack
- PoD attack
- Smurf attack
- Pulse wave attack
- Zero-day attack
- SYN flood attack
- Fragmentation attack
- ACK flood attack
- TCP state exhaustion attack
- Spoofed session flood attack
- HTTPS GET/POST attack
- Slowloris attack
- UDP application layer flood attack
- Multi-vector attack
- Peer-to-peer attack
- Permanent DoS (PDoS) attack
- Distributed reflection DoS (DRDoS) attack



UDP Flood Attack

- An attacker sends **spoofed UDP packets** at a very high packet rate to a remote host on random ports of a target server using a large source IP range
- The flooding of UDP packets causes the server to repeatedly check for **non-existent applications** at the ports
- Legitimate applications are inaccessible by the system and give an **error reply** with an ICMP "Destination Unreachable" packet
- This attack consumes **network resources** and available bandwidth, exhausting the network until it goes offline

Attacker
The attacker sends UDP packets with spoofed IP address and random destination UDP ports
Target Server
UDP Packet
UDP Packet
UDP Packet
UDP Packet
ICMP error packets of destination unreachable

UDP Flood Attack

In a UDP flood attack, an attacker sends spoofed UDP packets at a very high packet rate to a remote host on random ports of a target server by using a large source IP range. The flooding of UDP packets causes the server to check repeatedly for nonexistent applications at the ports. Consequently, legitimate applications become inaccessible by the system, and any attempts to access them return an error reply with an ICMP "Destination Unreachable" packet. This attack consumes network resources and available bandwidth, exhausting the network until it goes offline.

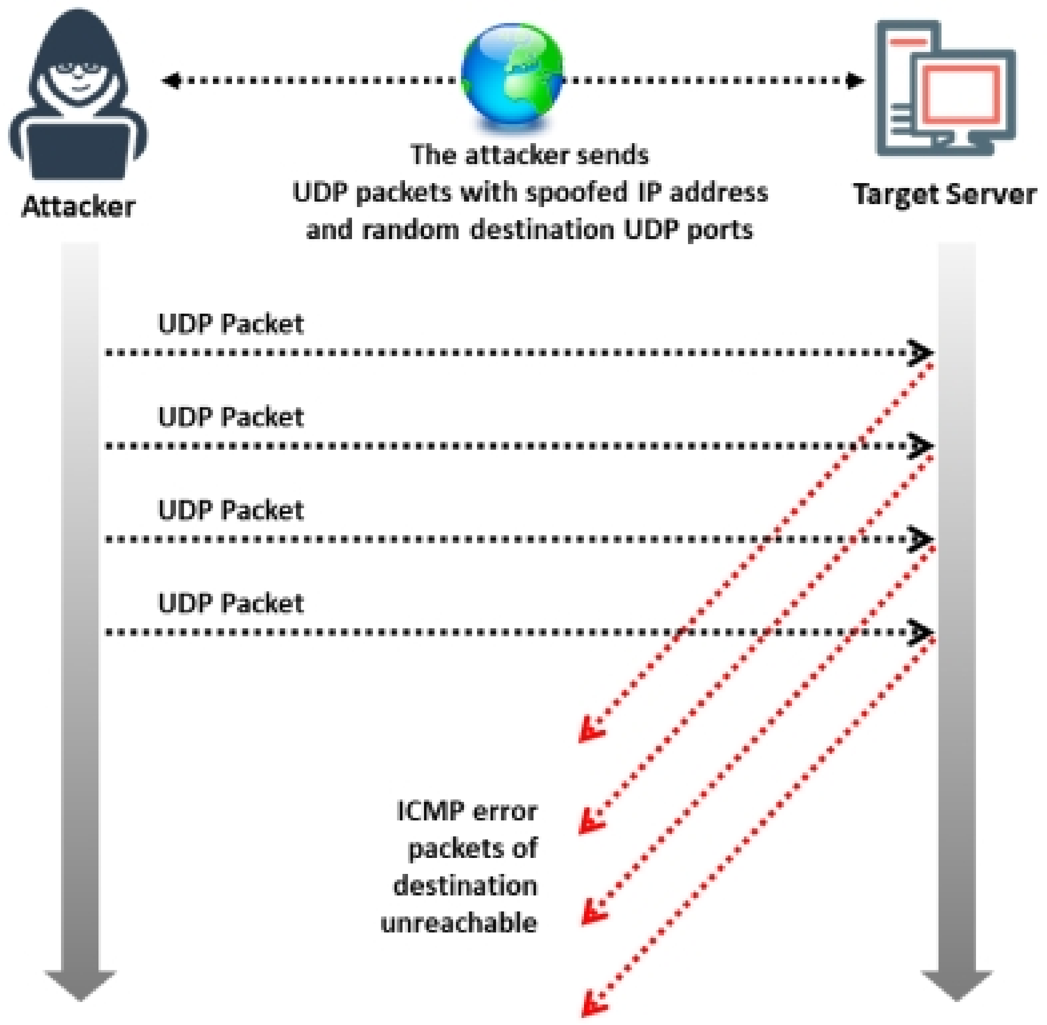
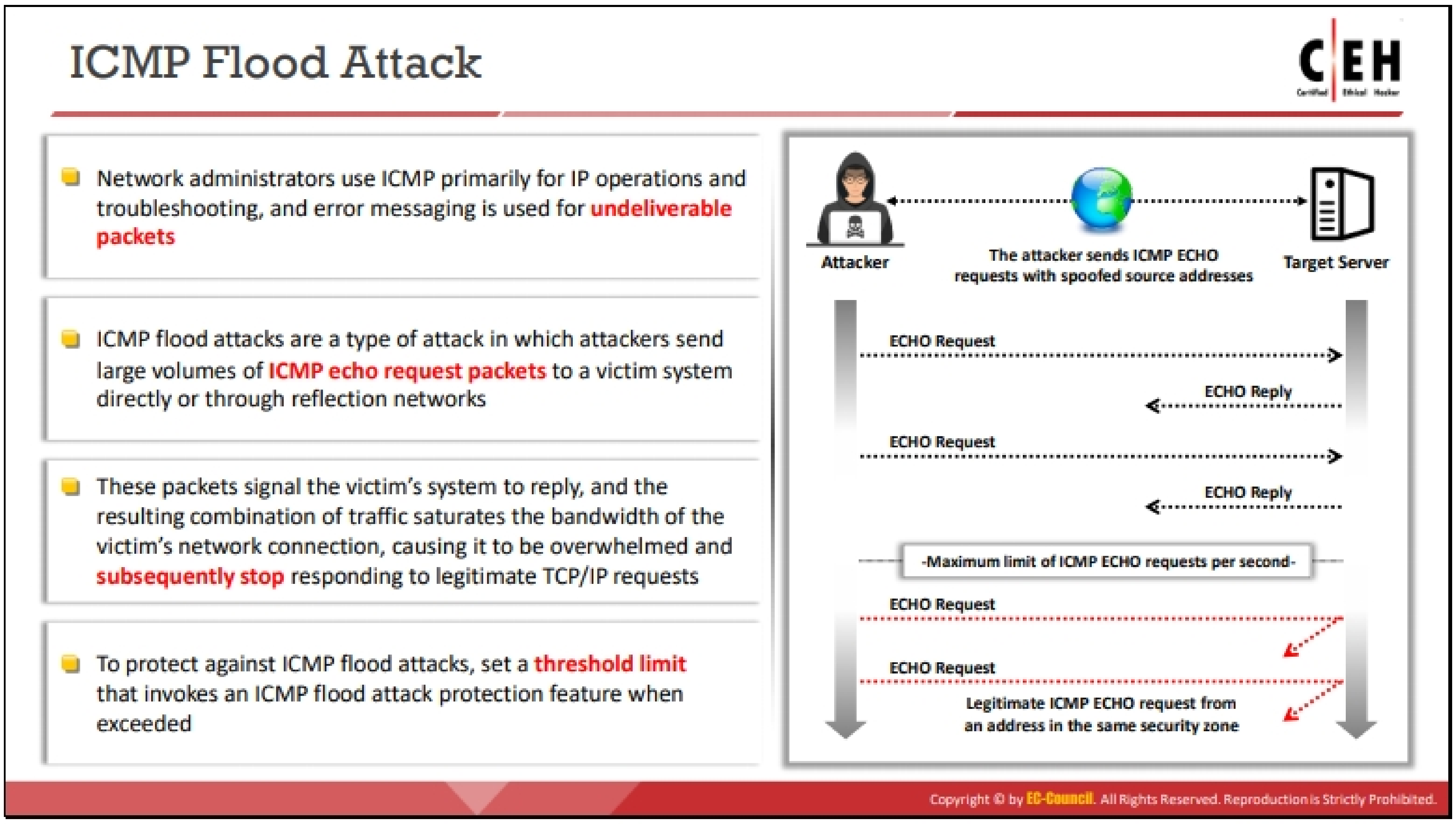

Attacker
The attacker sends UDP packets with spoofed IP address and random destination UDP ports
Target Server
UDP Packet
UDP Packet
UDP Packet
UDP Packet
ICMP error packets of destination unreachable

Figure 10.3: UDP flood attack



ICMP Flood Attack

- Network administrators use ICMP primarily for IP operations and troubleshooting, and error messaging is used for **undeliverable packets**
- ICMP flood attacks are a type of attack in which attackers send large volumes of **ICMP echo request packets** to a victim system directly or through reflection networks
- These packets signal the victim's system to reply, and the resulting combination of traffic saturates the bandwidth of the victim's network connection, causing it to be overwhelmed and **subsequently stop** responding to legitimate TCP/IP requests
- To protect against ICMP flood attacks, set a **threshold limit** that invokes an ICMP flood attack protection feature when exceeded

The attacker sends ICMP ECHO requests with spoofed source addresses

Legitimate ICMP ECHO request from an address in the same security zone

ICMP Flood Attack

Network administrators use ICMP primarily for IP operations, troubleshooting, and error messaging for undeliverable packets. In this attack, attackers send large volumes of ICMP echo request packets to a victim's system directly or through reflection networks. These packets signal the victim's system to reply, and the large traffic saturates the bandwidth of the victim's network connection, causing it to be overwhelmed and subsequently stop responding to legitimate TCP/IP requests.

To protect against ICMP flood attacks, it is necessary to set a threshold that invokes the ICMP flood attack protection feature when exceeded. When the ICMP threshold is exceeded (by default, the threshold value is 1000 packets/s), the router rejects further ICMP echo requests from all addresses in the same security zone for the remainder of the current second as well as the next second.

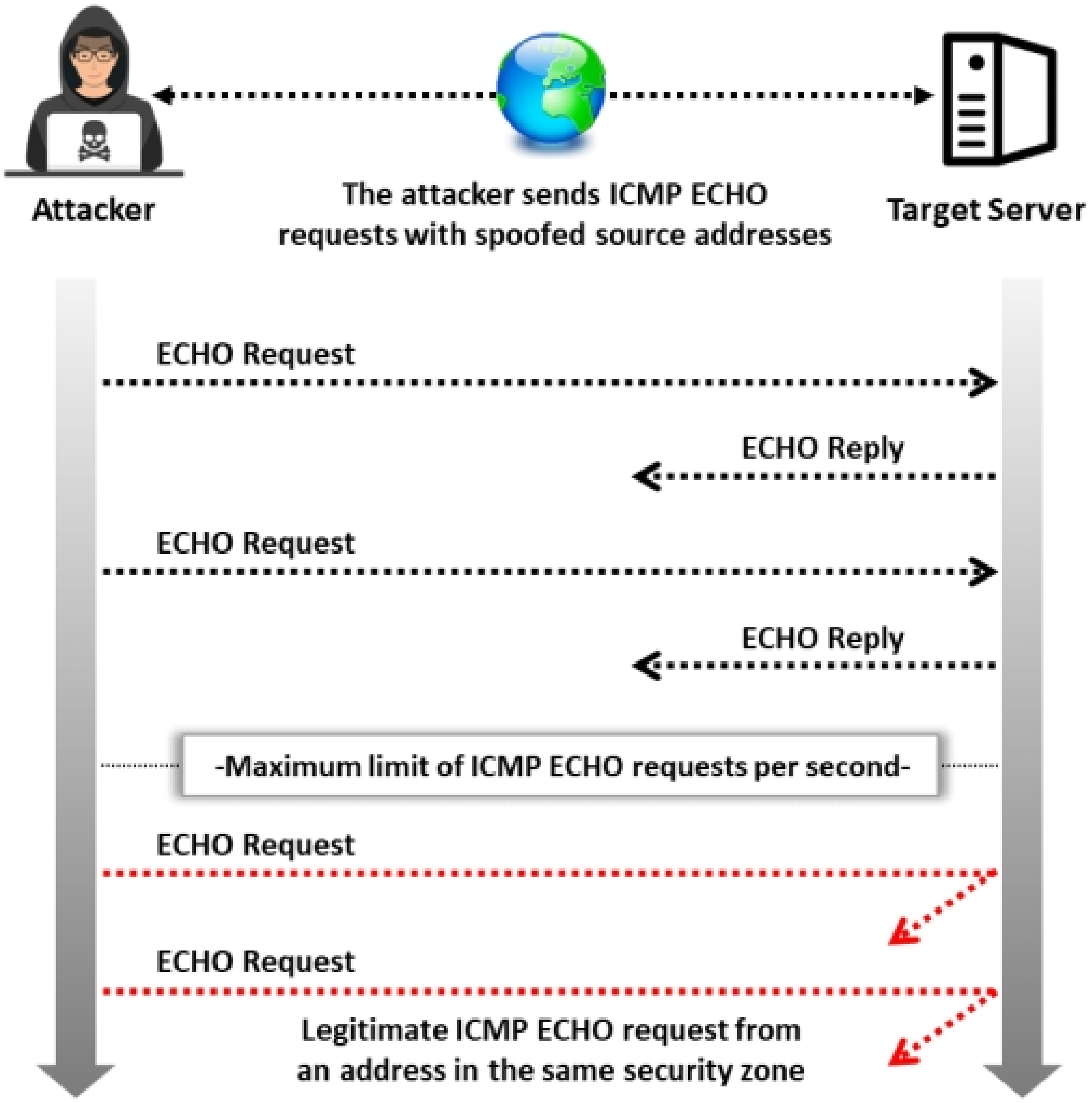
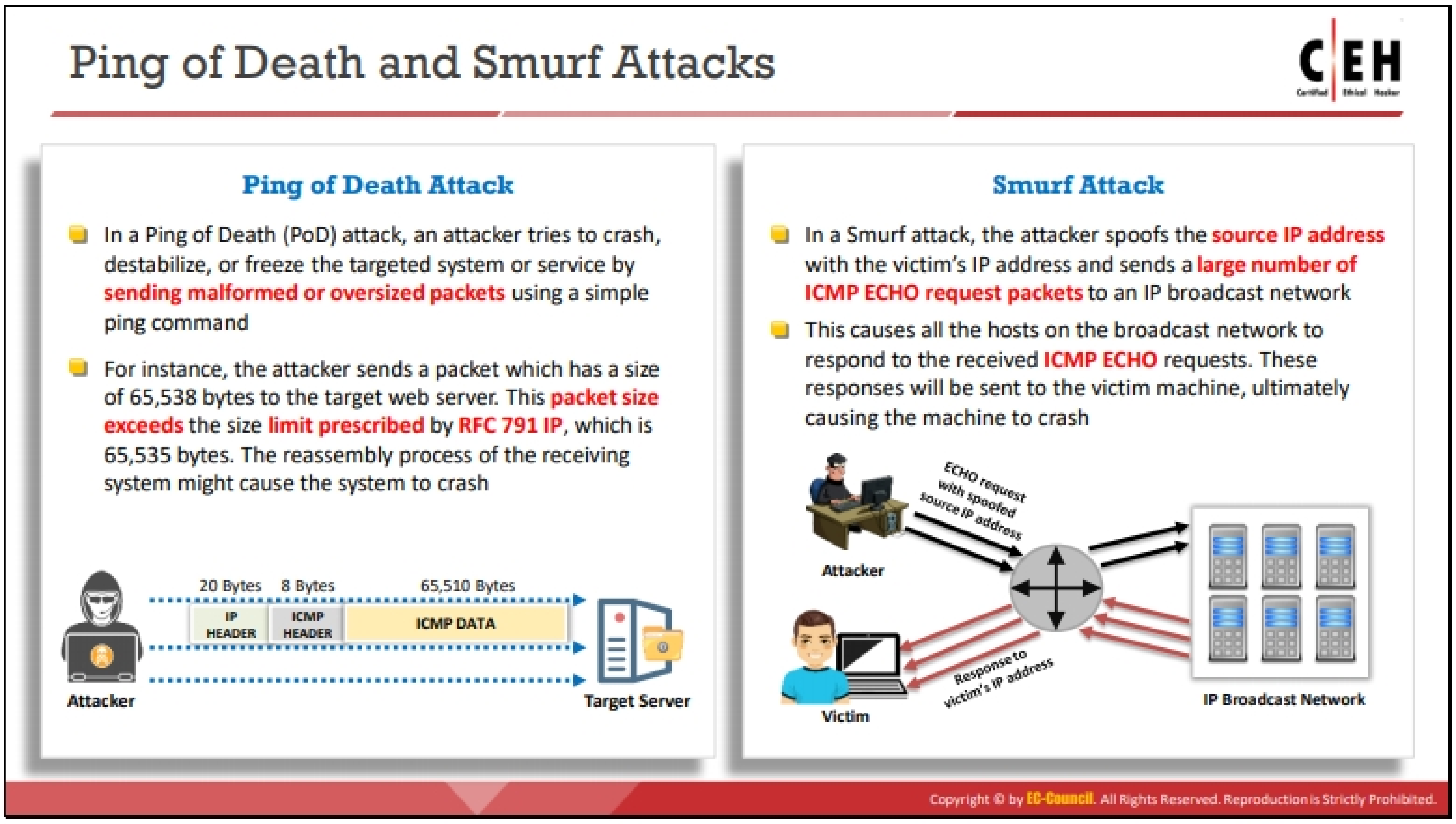

Attacker
The attacker sends ICMP ECHO requests with spoofed source addresses
Target Server
ECHO Request
ECHO Reply
ECHO Request
ECHO Reply
-Maximum limit of ICMP ECHO requests per second-
ECHO Request
ECHO Request
Legitimate ICMP ECHO request from an address in the same security zone

Figure 10.4: ICMP flood attack



Ping of Death and Smurf Attacks

Ping of Death Attack

- In a Ping of Death (PoD) attack, an attacker tries to crash, destabilize, or freeze the targeted system or service by **sending malformed or oversized packets** using a simple ping command
- For instance, the attacker sends a packet which has a size of 65,538 bytes to the target web server. This **packet size exceeds** the size **limit prescribed** by **RFC 791 IP**, which is 65,535 bytes. The reassembly process of the receiving system might cause the system to crash

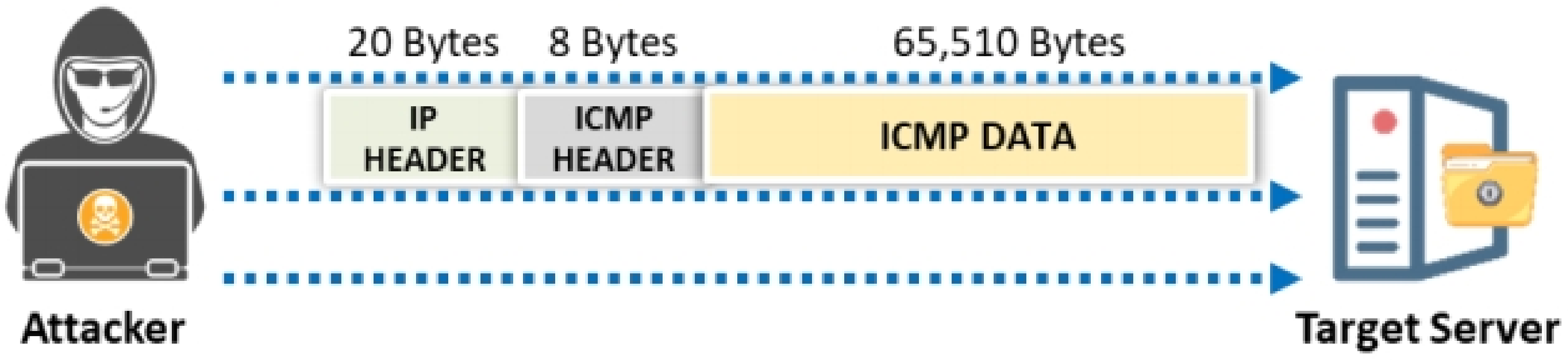
Smurf Attack

- In a Smurf attack, the attacker spoofs the **source IP address** with the victim's IP address and sends a **large number of ICMP ECHO request packets** to an IP broadcast network
- This causes all the hosts on the broadcast network to respond to the received **ICMP ECHO** requests. These responses will be sent to the victim machine, ultimately causing the machine to crash

20 Bytes
8 Bytes
65,510 Bytes
IP HEADER
ICMP HEADER
ICMP DATA
Attacker
Target Server
ECHO request with spoofed source IP address
Attacker
Victim
Response to victim's IP address
IP Broadcast Network

Ping of Death Attack

In a Ping of Death (PoD) attack, an attacker attempts to crash, destabilize, or freeze the target system or service by sending malformed or oversized packets using a simple ping command. Suppose an attacker sends a packet with a size of 65,538 bytes to the target web server. This size exceeds the size limit prescribed by RFC 791 IP, which is 65,535 bytes. The reassembly process performed by the receiving system might cause the system to crash. In such attacks, the attacker's identity can be easily spoofed, and the attacker might not need detailed knowledge of the target machine, except its IP address.



20 Bytes
8 Bytes
65,510 Bytes
IP HEADER
ICMP HEADER
ICMP DATA
Attacker
Target Server

Figure 10.5: Ping-of-death attack

Smurf Attack

In a Smurf attack, the attacker spoofs the source IP address with the victim's IP address and sends a large number of ICMP ECHO request packets to an IP broadcast network. This causes all the hosts on the broadcast network to respond to the received ICMP ECHO requests. These responses are sent to the victim's machine because the IP address was spoofed by the attacker, causing significant traffic to the victim's machine and ultimately making it crash.

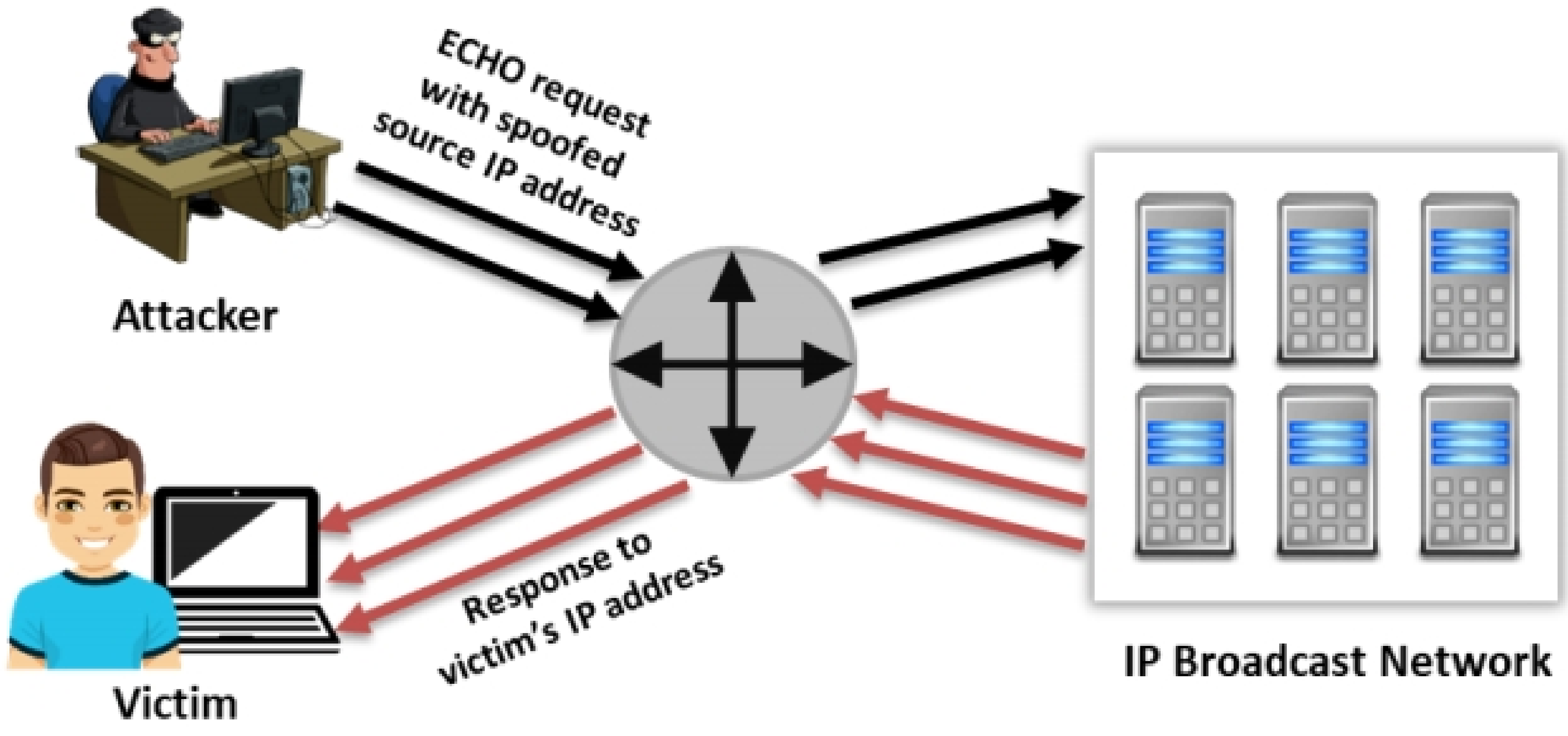
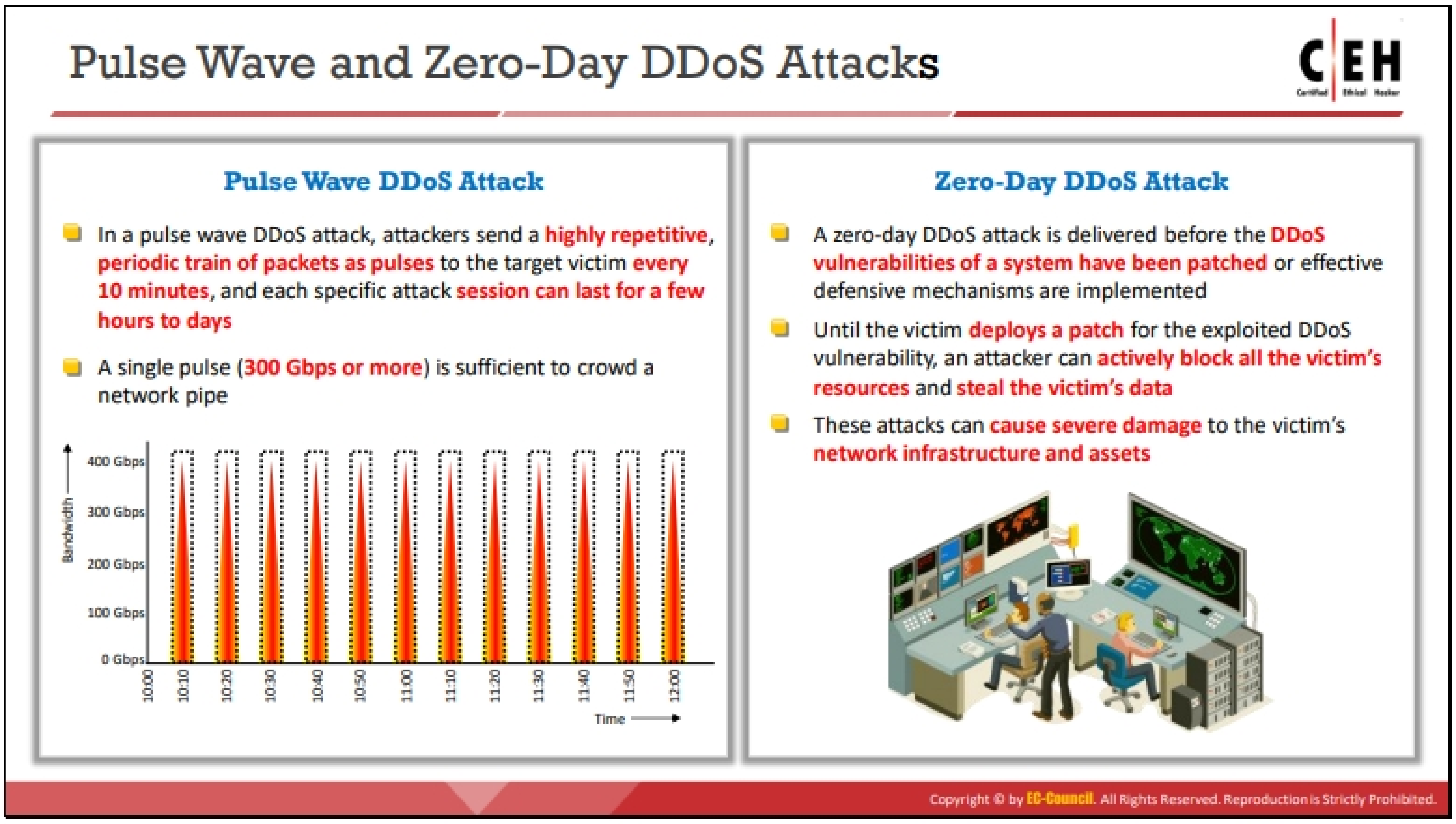

ECHO request with spoofed source IP address
Attacker
Victim
Response to victim's IP address
IP Broadcast Network

Figure 10.6: Smurf attack



Pulse Wave and Zero-Day DDoS Attacks

Pulse Wave DDoS Attack

- In a pulse wave DDoS attack, attackers send a **highly repetitive, periodic train of packets as pulses** to the target victim **every 10 minutes**, and each specific attack **session can last for a few hours to days**
- A single pulse (**300 Gbps or more**) is sufficient to crowd a network pipe

Bandwidth
400 Gbps
300 Gbps
200 Gbps
100 Gbps
0 Gbps
10:00
10:10
10:20
10:30
10:40
10:50
11:00
11:10
11:20
11:30
11:40
11:50
12:00
Time

Zero-Day DDoS Attack

- A zero-day DDoS attack is delivered before the **DDoS vulnerabilities of a system have been patched** or effective defensive mechanisms are implemented
- Until the victim **deploys a patch** for the exploited DDoS vulnerability, an attacker can **actively block all the victim's resources** and **steal the victim's data**
- These attacks can **cause severe damage** to the victim's **network infrastructure and assets**

Pulse Wave DDoS Attack

Pulse wave DDoS attacks are the latest type of DDoS attacks employed by threat actors to disrupt the standard operations of targets. Generally, DDoS attack patterns are continuous incoming traffic flows. However, in pulse wave DDoS attacks, the attack pattern is periodic, and the attack is huge, consuming the entire bandwidth of target networks. Attackers send a highly repetitive train of packets as pulses to the target victim every 10 min, and the attack session lasts for approximately an hour or some days. A single pulse (300 Gbps or more) is more than enough to crowd a network pipe. Recovery from such attacks is very difficult and occasionally impossible.

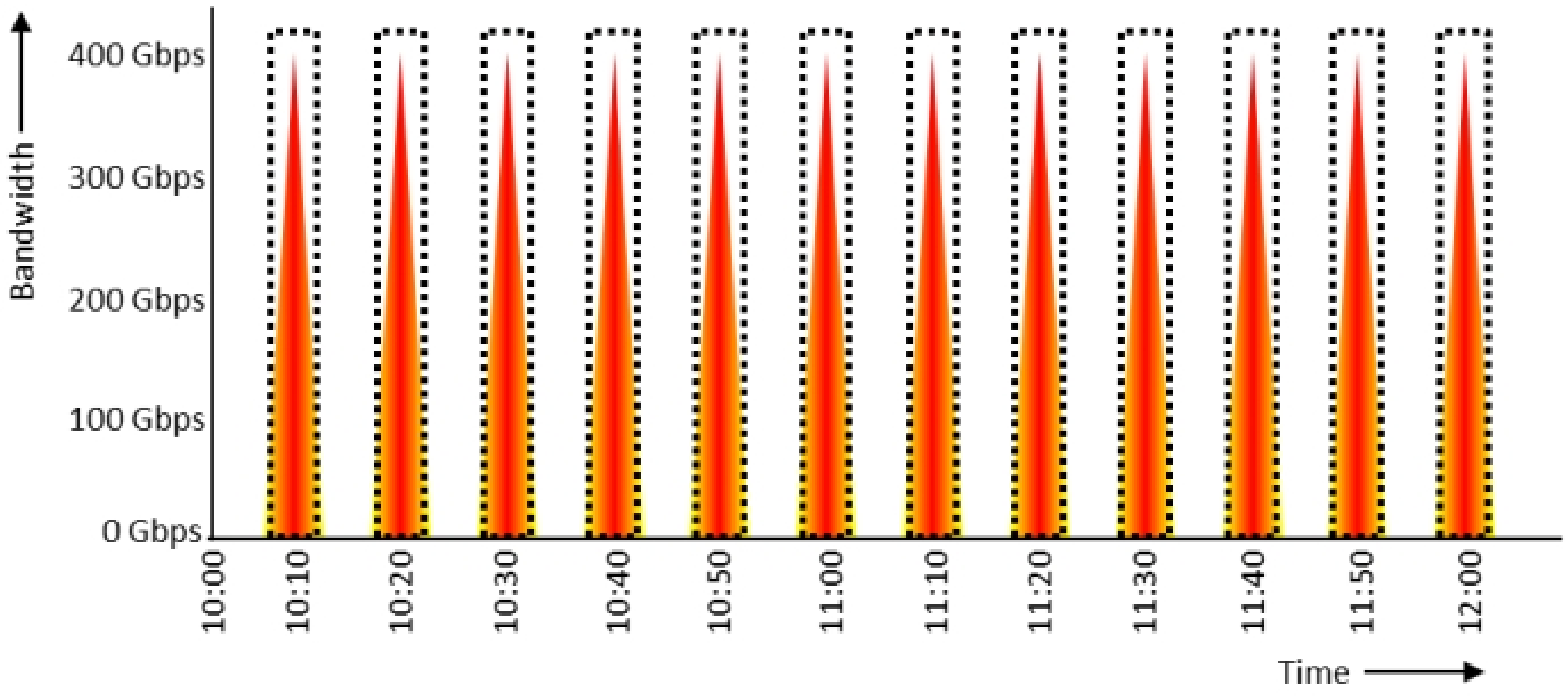
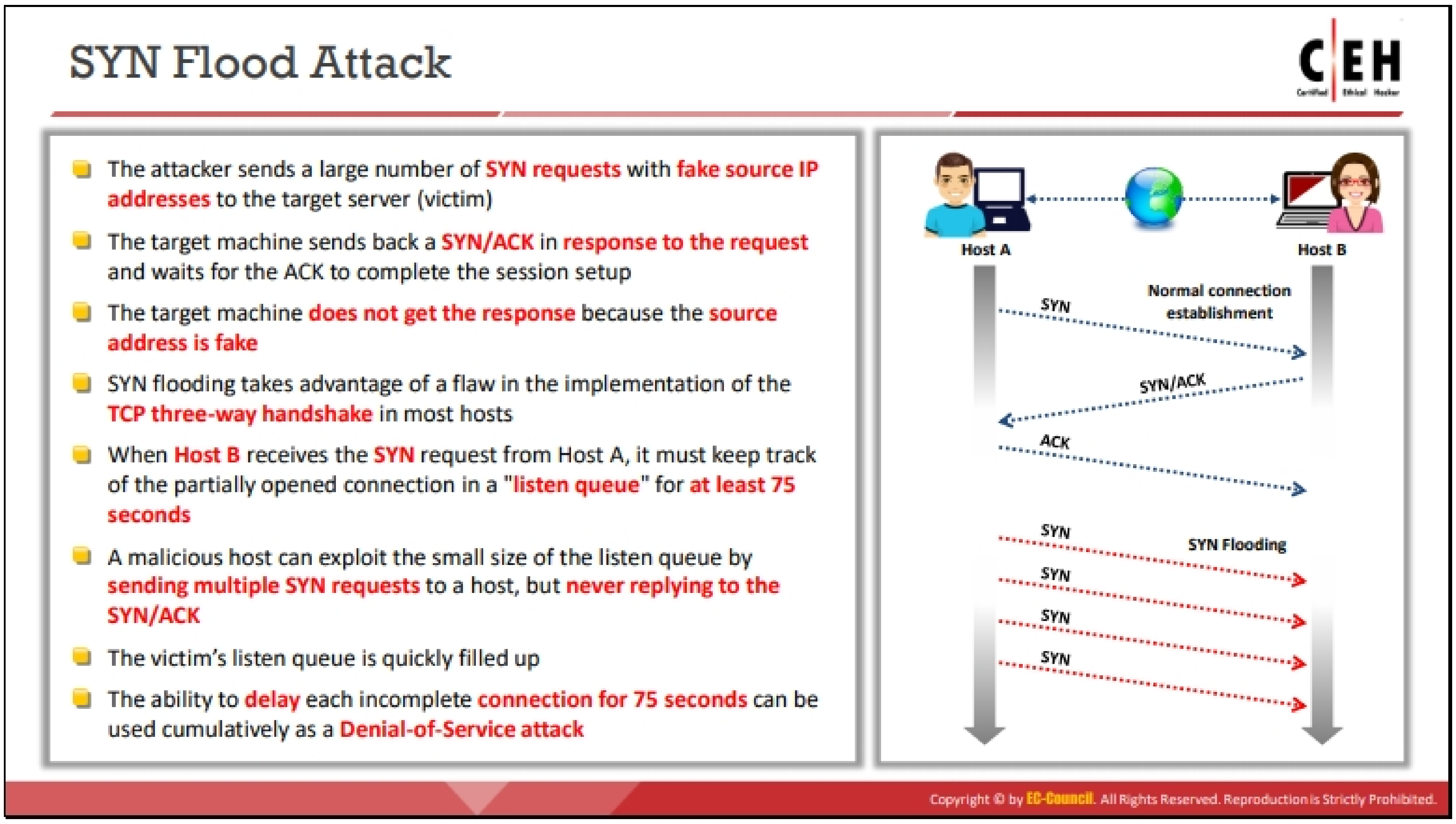


Figure 10.7: Pulse wave DDoS attack

Zero-Day DDoS Attack

Zero-day DDoS attacks are attacks in which DDoS vulnerabilities do not have patches or effective defensive mechanisms. Until the victim identifies the threat actor's attack strategy and deploys a patch for the exploited DDoS vulnerability, the attacker actively blocks all the victim's resources and steals the victim's data. These attacks can cause severe damage to the victim's network infrastructure and assets. Currently, there is no versatile approach to protect networks from this type of attack.



SYN Flood Attack

- The attacker sends a large number of **SYN requests** with **fake source IP addresses** to the target server (victim)
- The target machine sends back a **SYN/ACK** in **response to the request** and waits for the ACK to complete the session setup
- The target machine **does not get the response** because the **source address is fake**
- SYN flooding takes advantage of a flaw in the implementation of the **TCP three-way handshake** in most hosts
- When **Host B** receives the **SYN** request from Host A, it must keep track of the partially opened connection in a "**listen queue**" for **at least 75 seconds**
- A malicious host can exploit the small size of the listen queue by **sending multiple SYN requests** to a host, but **never replying to the SYN/ACK**
- The victim's listen queue is quickly filled up
- The ability to **delay** each incomplete **connection for 75 seconds** can be used cumulatively as a **Denial-of-Service attack**

SYN Flood Attack

In a SYN attack, the attacker sends a large number of SYN requests to the target server (victim) with fake source IP addresses. The attack creates incomplete TCP connections that use up network resources. Normally, when a client wants to begin a TCP connection to a server, the client and server exchange the following series of messages:

- A TCP SYN request packet is sent to a server.
- The server sends a SYN/ACK (acknowledgement) in response to the request.
- The client sends a response ACK to the server to complete the session setup.

This method is a "three-way handshake."

In a SYN attack, the attacker exploits the three-way handshake method. First, the attacker sends a fake TCP SYN request to the target server. After the server sends a SYN/ACK in response to the client's (attacker's) request, the client never sends an ACK response. This leaves the server waiting to complete the connection.

SYN flooding takes advantage of the flawed manner in which most hosts implement the TCP three-way handshake. This attack occurs when the attacker sends unlimited SYN packets (requests) to the host system. The process of transmitting such packets is faster than the system can handle. Normally, a connection is established with the TCP three-way handshake. The host keeps track of partially open connections while waiting for response ACK packets in a listening queue.

As shown in the figure, when Host B receives a SYN request from Host A, it must keep track of the partially opened connection in a "listen queue" for at least 75 s.

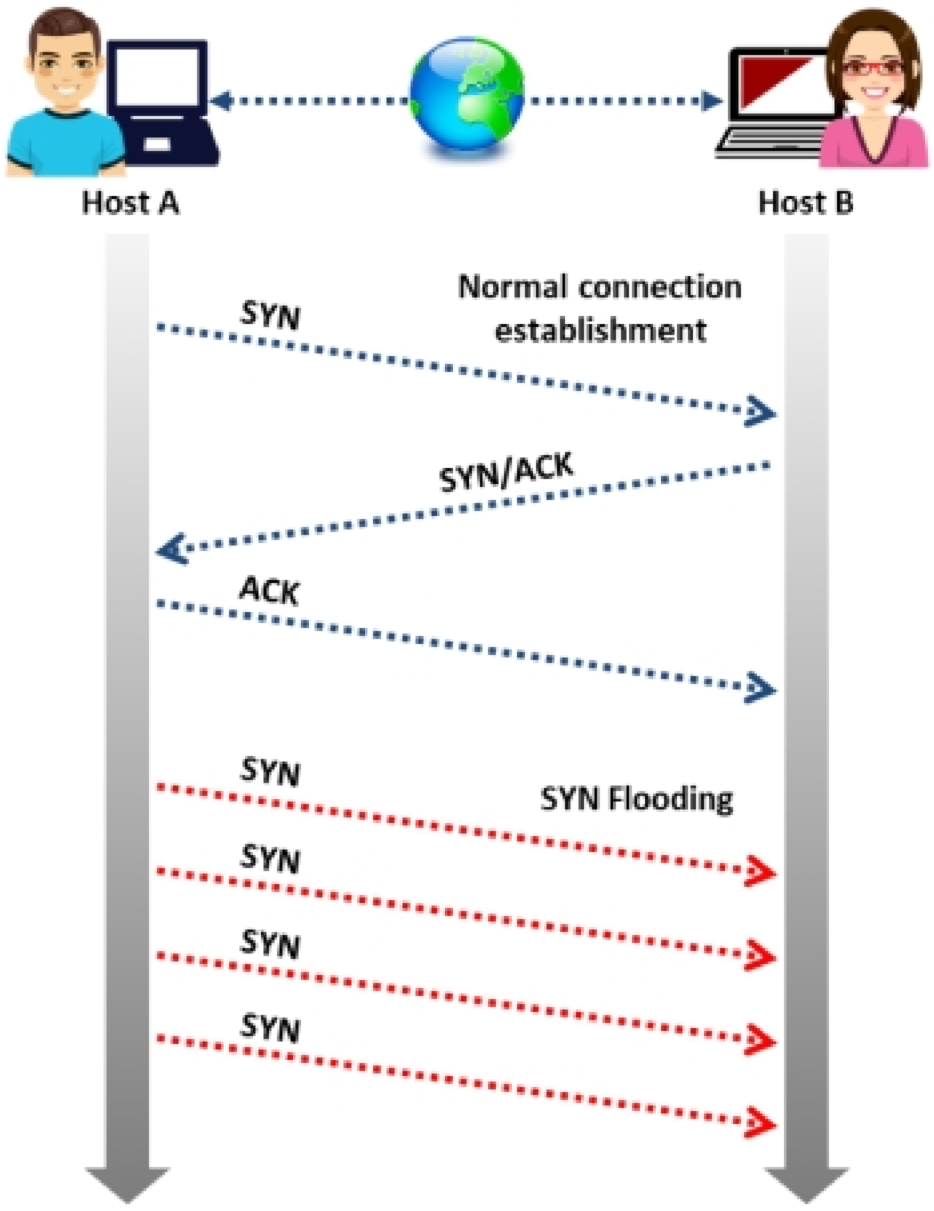


Figure 10.8: SYN flood attack

A malicious host can exploit another host, managing many partial connections by sending many SYN requests to the target host simultaneously. When the queue is full, the system cannot open new connections until it drops some entries from the connection queue through handshake timeouts. This ability to hold up each incomplete connection for 75 s can be cumulatively exploited in a DoS attack. The attack uses fake IP addresses, making it difficult to trace the source. An attacker can fill a table of connections even without spoofing the source IP address.

In addition to SYN flood attacks, attackers can also employ SYN-ACK and ACK/PUSH ACK flood attacks to disrupt target machines. All these attacks are similar in functionality with minor variations.

SYN-ACK Flood Attack

This type of attack is similar to the SYN flood attack, except that in this type of flood attack, the attacker exploits the second stage of a three-way handshake by sending a large number of SYN-ACK packets to the target machine to exhaust its resources.

ACK and PUSH ACK Flood Attack

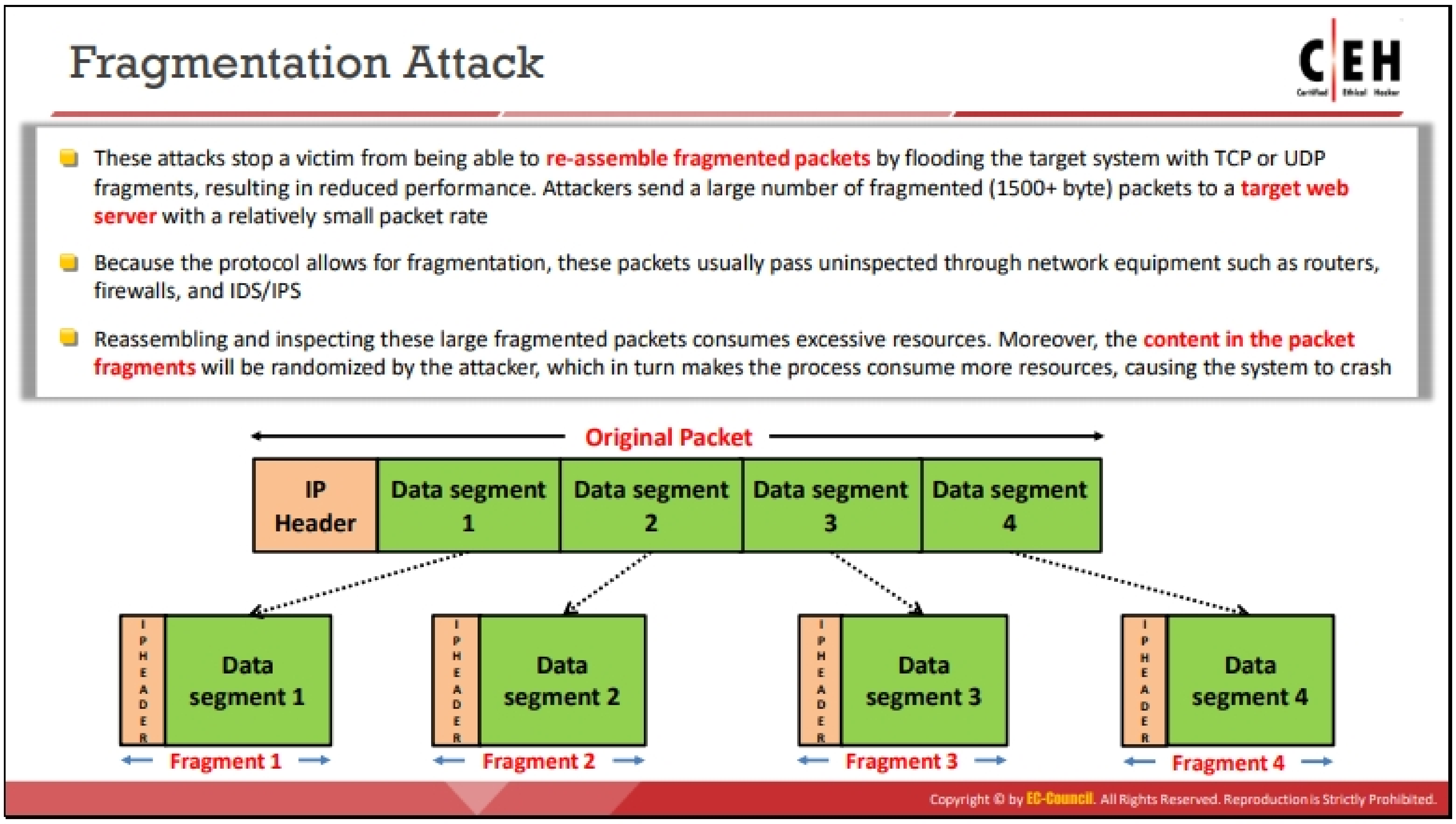
During an active TCP session, ACK and PUSH ACK are the flags used to transfer information to and from the server and client machines till the session ends. In an ACK and PUSH ACK flood attack, attackers send a large amount of spoofed ACK and PUSH ACK packets to the target machine, making it non-functional.

Countermeasures for SYN Flood Attacks

Proper packet filtering is a viable solution to SYN flood attacks. An administrator can also tune the TCP/IP stack to reduce the impact of SYN attacks while allowing legitimate client traffic.

Some SYN attacks do not attempt to upset servers; instead, they attempt to consume the entire bandwidth of the Internet connection. Two tools to counter this attack are SYN cookies and SynAttackProtect.

To guard against an attacker attempting to consume the bandwidth of an Internet connection, an administrator can implement some additional safety measures; for example, they can decrease the time-out period in which a pending connection is maintained in the "SYN RECEIVED" state in the queue. Normally, if a client sends no response ACK, a server will retransmit the first ACK packet. This vulnerability can be removed by decreasing the time of the first packet's retransmission, decreasing the number of packet retransmissions, or turning off packet retransmissions entirely.



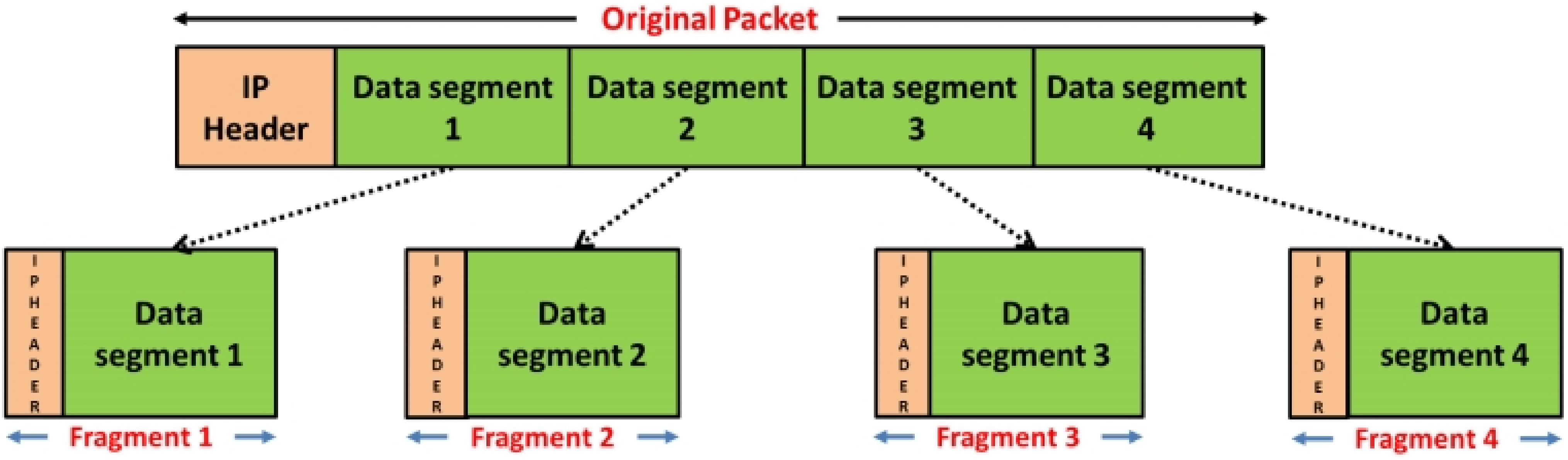
Fragmentation Attack

- These attacks stop a victim from being able to re-assemble fragmented packets by flooding the target system with TCP or UDP fragments, resulting in reduced performance. Attackers send a large number of fragmented (1500+ byte) packets to a target web server with a relatively small packet rate
- Because the protocol allows for fragmentation, these packets usually pass uninspected through network equipment such as routers, firewalls, and IDS/IPS
- Reassembling and inspecting these large fragmented packets consumes excessive resources. Moreover, the content in the packet fragments will be randomized by the attacker, which in turn makes the process consume more resources, causing the system to crash

Original Packet
IP Header
Data segment 1
Data segment 2
Data segment 3
Data segment 4
IPHEADER
Data segment 1
Fragment 1
IPHEADER
Data segment 2
Fragment 2
IPHEADER
Data segment 3
Fragment 3
IPHEADER
Data segment 4
Fragment 4

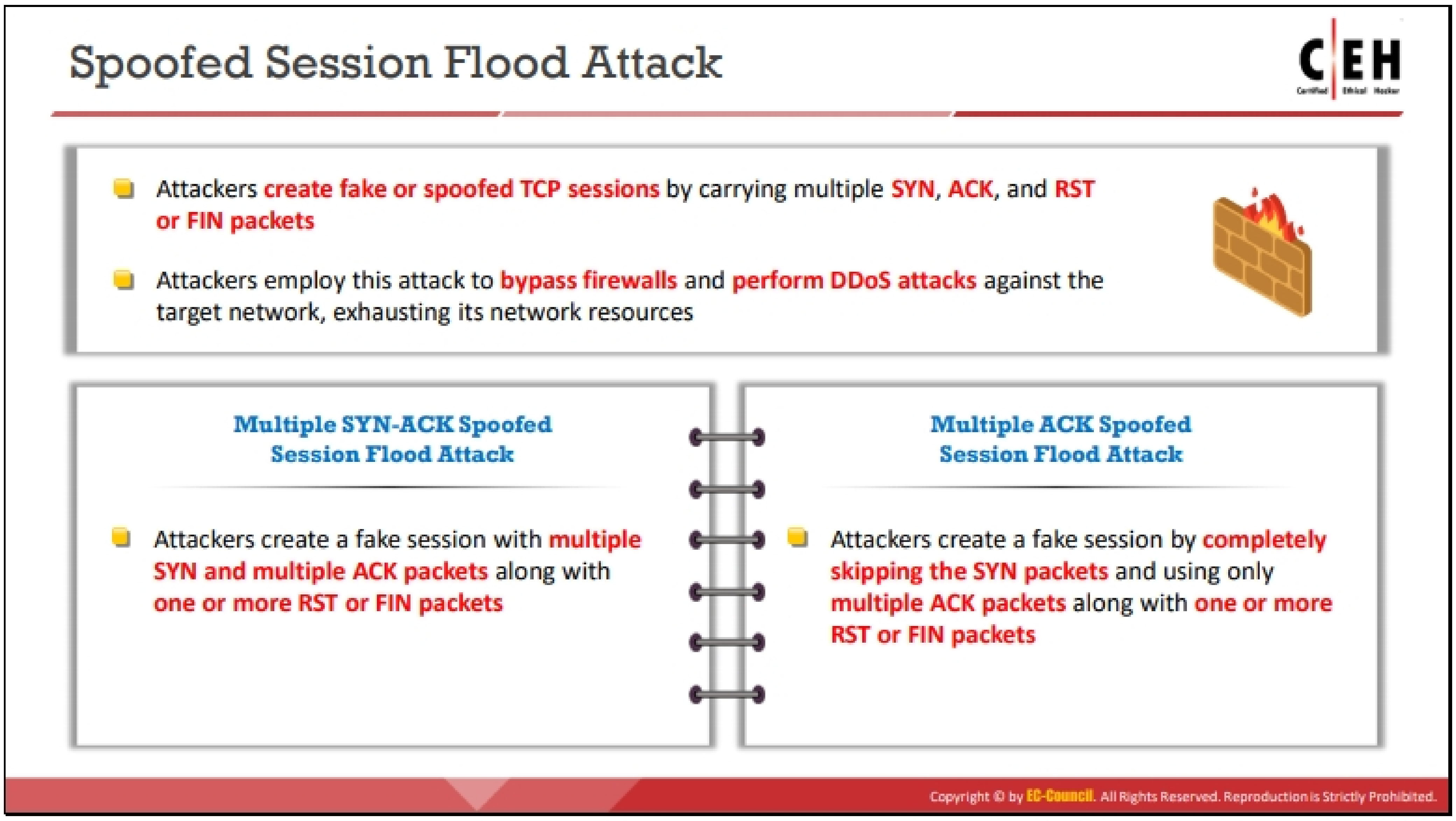
Fragmentation Attack

These attacks destroy a victim's ability to reassemble fragmented packets by flooding it with TCP or UDP fragments, resulting in reduced performance. In fragmentation attacks, the attacker sends a large number of fragmented (1500+ byte) packets to a target web server with a relatively small packet rate. Since the protocol allows fragmentation, these packets are usually uninspected as they pass through network equipment such as routers, firewalls, and the intrusion detection system (IDS)/intrusion prevention system (IPS). The reassembly and inspection of these large fragmented packets consume excessive resources. Moreover, the content in the packet fragments is randomized by the attacker, which makes the reassembly and inspection consume more resources and, in turn, causes the system to crash.



Original Packet
IP Header
Data segment 1
Data segment 2
Data segment 3
Data segment 4
IPHEADER
Data segment 1
Fragment 1
IPHEADER
Data segment 2
Fragment 2
IPHEADER
Data segment 3
Fragment 3
IPHEADER
Data segment 4
Fragment 4

Figure 10.9: Fragmentation attack



Spoofed Session Flood Attack

- Attackers **create fake or spoofed TCP sessions** by carrying multiple **SYN**, **ACK**, and **RST or FIN packets**
- Attackers employ this attack to **bypass firewalls** and **perform DDoS attacks** against the target network, exhausting its network resources

Multiple SYN-ACK Spoofed Session Flood Attack

- Attackers create a fake session with **multiple SYN and multiple ACK packets** along with **one or more RST or FIN packets**

Multiple ACK Spoofed Session Flood Attack

- Attackers create a fake session by **completely skipping the SYN packets** and using only **multiple ACK packets** along with **one or more RST or FIN packets**

Spoofed Session Flood Attack

In this type of attack, attackers create fake or spoofed TCP sessions by carrying multiple SYN, ACK, and RST or FIN packets. Attackers employ this attack to bypass firewalls and perform DDoS attacks against target networks, exhausting their network resources.

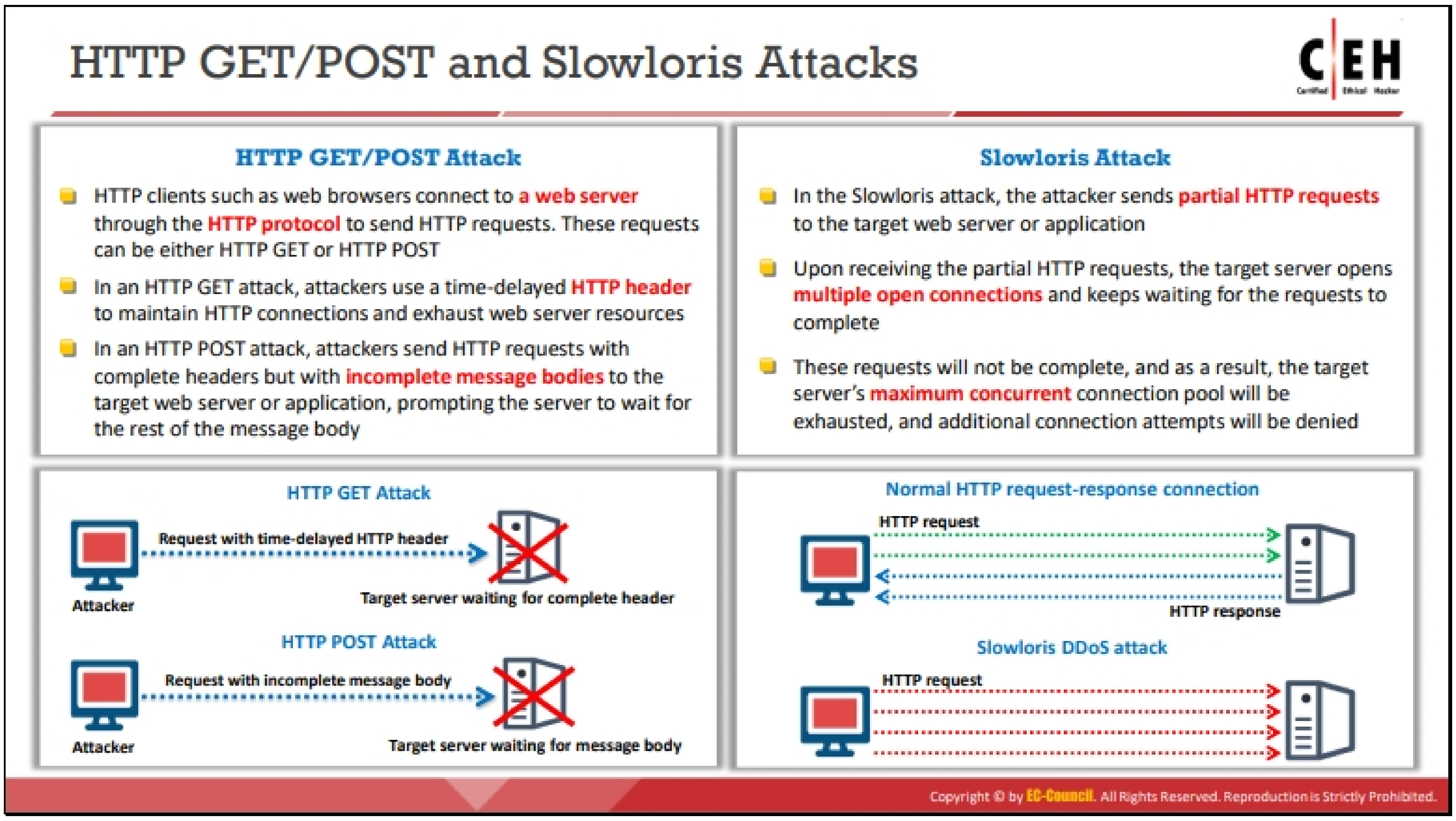
The following are examples for spoofed session flood attacks:

- **Multiple SYN-ACK Spoofed Session Flood Attack**

  In this type of flood attack, attackers create a fake session with multiple SYN and multiple ACK packets, along with one or more RST or FIN packets.

- **Multiple ACK Spoofed Session Flood Attack**

  In this type of flood attack, attackers create a fake session by completely skipping SYN packets and using only multiple ACK packets along with one or more RST or FIN packets. Because SYN packets are not employed and firewalls mostly use SYN packet filters to detect abnormal traffic, the DDoS detection rate of the firewalls is very low for these types of attacks.



HTTP GET/POST and Slowloris Attacks

HTTP GET/POST Attack

- HTTP clients such as web browsers connect to **a web server** through the **HTTP protocol** to send HTTP requests. These requests can be either HTTP GET or HTTP POST
- In an HTTP GET attack, attackers use a time-delayed **HTTP header** to maintain HTTP connections and exhaust web server resources
- In an HTTP POST attack, attackers send HTTP requests with complete headers but with **incomplete message bodies** to the target web server or application, prompting the server to wait for the rest of the message body

Slowloris Attack

- In the Slowloris attack, the attacker sends **partial HTTP requests** to the target web server or application
- Upon receiving the partial HTTP requests, the target server opens **multiple open connections** and keeps waiting for the requests to complete
- These requests will not be complete, and as a result, the target server's **maximum concurrent** connection pool will be exhausted, and additional connection attempts will be denied

HTTP GET Attack
Request with time-delayed HTTP header
Attacker
Target server waiting for complete header
HTTP POST Attack
Request with incomplete message body
Attacker
Target server waiting for message body
Normal HTTP request-response connection
HTTP request
HTTP response
Slowloris DDoS attack
HTTP request

HTTP GET/POST Attack

HTTP attacks are layer-7 attacks. HTTP clients, such as web browsers, connect to a web server through HTTP to send HTTP requests, which can be either HTTP GET or HTTP POST. Attackers exploit these requests to perform DoS attacks.

In an HTTP GET attack, the attacker uses a time-delayed HTTP header to hold on to an HTTP connection and exhaust web-server resources. The attacker never sends the full request to the target server. Consequently, the server retains the HTTP connection and waits, making it inaccessible for legitimate users. In these types of attacks, all the network parameters appear healthy while the service remains unavailable.

In an HTTP POST attack, the attacker sends HTTP requests with complete headers but an incomplete message body to the target web server or application. Because the message body is incomplete, the server waits for the rest of the body, making the web server or web application unavailable to legitimate users.

An HTTP GET/POST attack is a sophisticated layer-7 attack that does not use malformed packets, spoofing, or reflection techniques. This type of attack requires less bandwidth than other attacks to bring down the targeted site or web server. This attack aims to compel the server to allocate as many resources as possible to serve the attack, thereby denying legitimate users access to the server's resources.

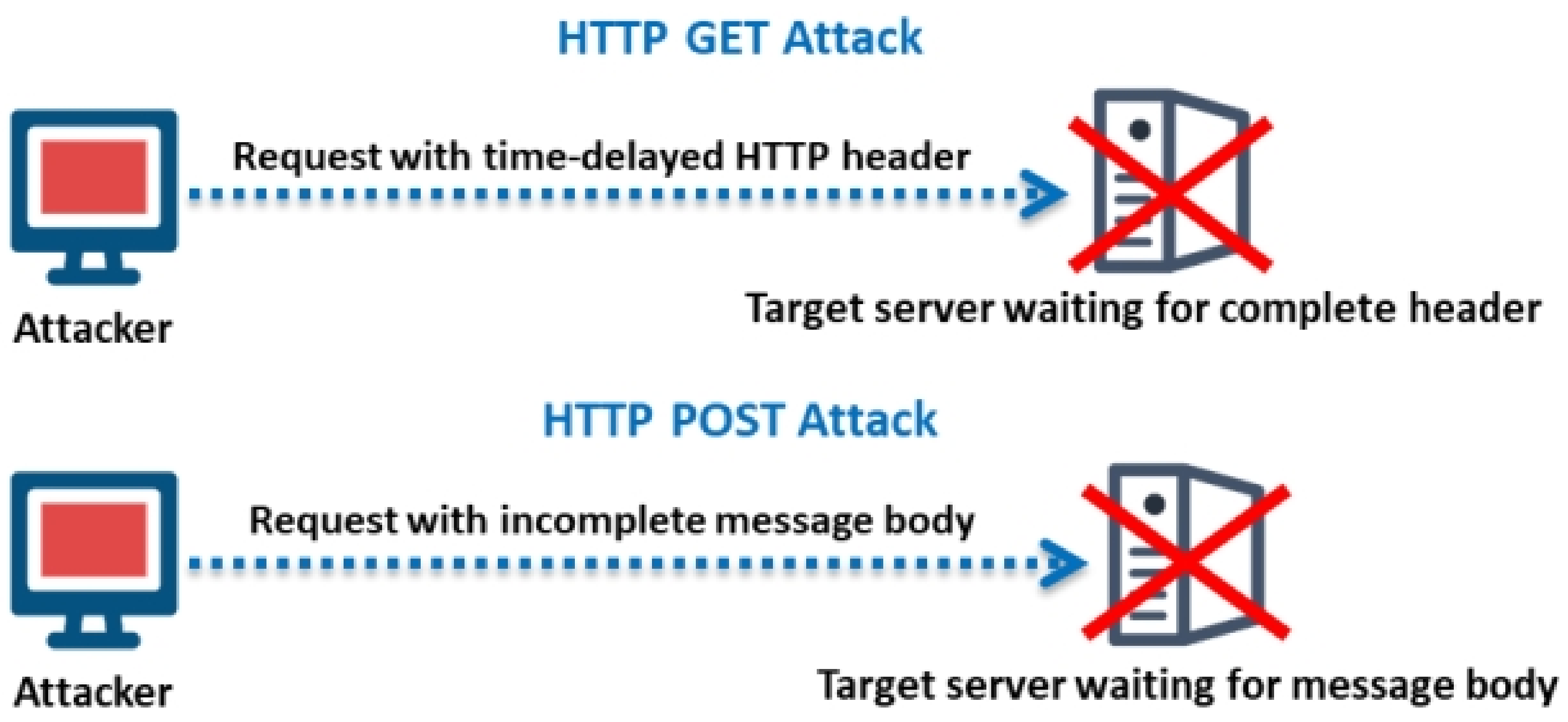

HTTP GET Attack
Request with time-delayed HTTP header
Attacker
Target server waiting for complete header
HTTP POST Attack
Request with incomplete message body
Attacker
Target server waiting for message body

Figure 10.10: HTTP GET/POST attack

In addition to the aforementioned HTTP GET/POST attack, attackers can employ the following HTTP flood attacks to exhaust the target network's bandwidth:

- **Single-Session HTTP Flood Attack**

  In this type of flood attack, an attacker exploits the vulnerabilities in HTTP 1.1 to bombard a target with multiple requests in a single HTTP session.

- **Single-Request HTTP Flood Attack**

  In this type of flood attack, attackers make several HTTP requests from a single HTTP session by masking these requests within one HTTP packet. This technique allows attackers to be anonymous and invisible while performing DDoS attacks.

- **Recursive HTTP GET Flood Attack**

  Staying undetected is key for attackers. An attacker posing as a legitimate user and performing legitimate actions can trick any firewall into believing that the source is legitimate while it is not. Recursive GET collects a list of pages or images and appears to be going through these pages or images. However, it stealthily performs flooding attacks on the target. The recursive GET in combination with an HTTP flood attack can cause extreme damage to the target.

- **Random Recursive GET Flood Attack**

  This type of attack is a tweaked version of the recursive GET flood attack. It is designed for forums, blogs, and other websites that have pages in a sequence. Similar to the recursive GET flood attack, in this attack, the recursive GET pretends to be going through pages. Because the targets are forums, groups, and other blogs, the attacker uses random numbers from a valid page range to pose as a legitimate user and sends a new GET request each time. In both recursive GET and random recursive GET flood attacks, the target is bombarded with a large number of GET requests, exhausting its resources.

Slowloris Attack

Slowloris is a DDoS attack tool used to perform layer-7 DDoS attacks to take down web infrastructure. It is distinctly different from other tools in that it uses perfectly legitimate HTTP traffic to take down a target server. In Slowloris attacks, the attacker sends partial HTTP requests to the target web server or application. Upon receiving the partial requests, the target server opens multiple connections and waits for the requests to complete. However, these requests remain incomplete, causing the target server's maximum concurrent connection pool to be filled up and additional connection attempts to be denied.

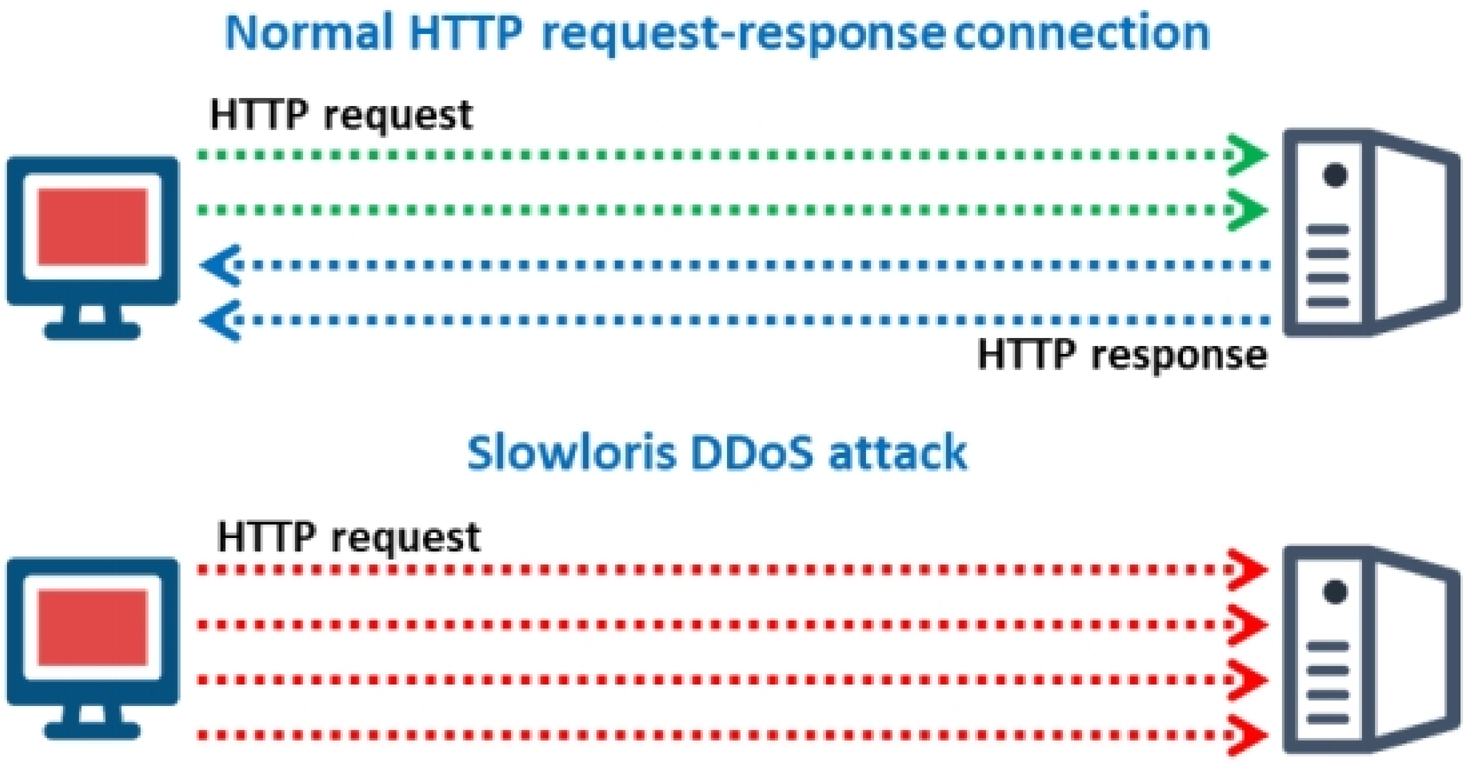
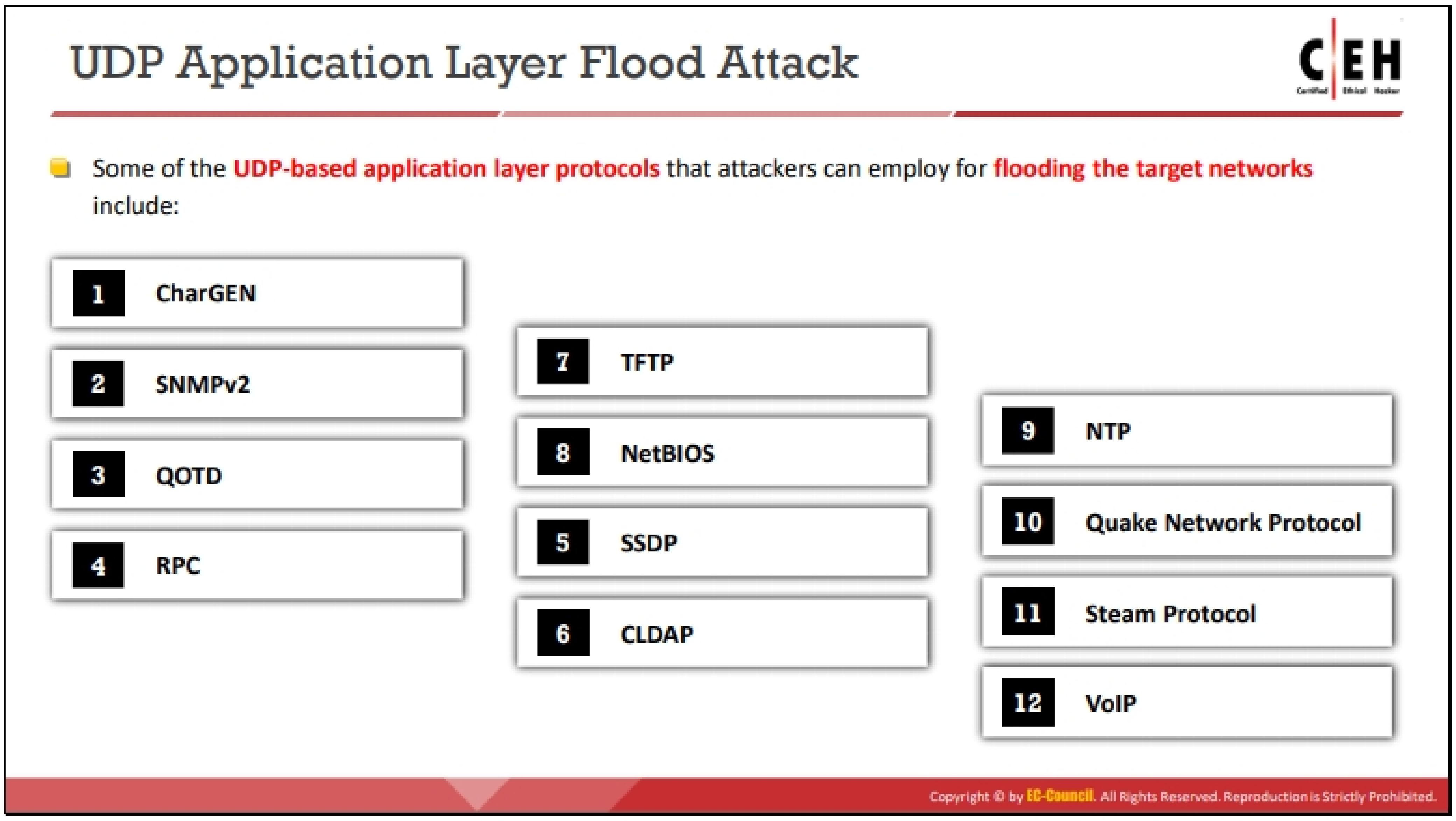

Normal HTTP request-response connection
HTTP request
HTTP response
Slowloris DDoS attack
HTTP request

Figure 10.11: Slowloris attack



UDP Application Layer Flood Attack

- Some of the **UDP-based application layer protocols** that attackers can employ for **flooding the target networks** include:

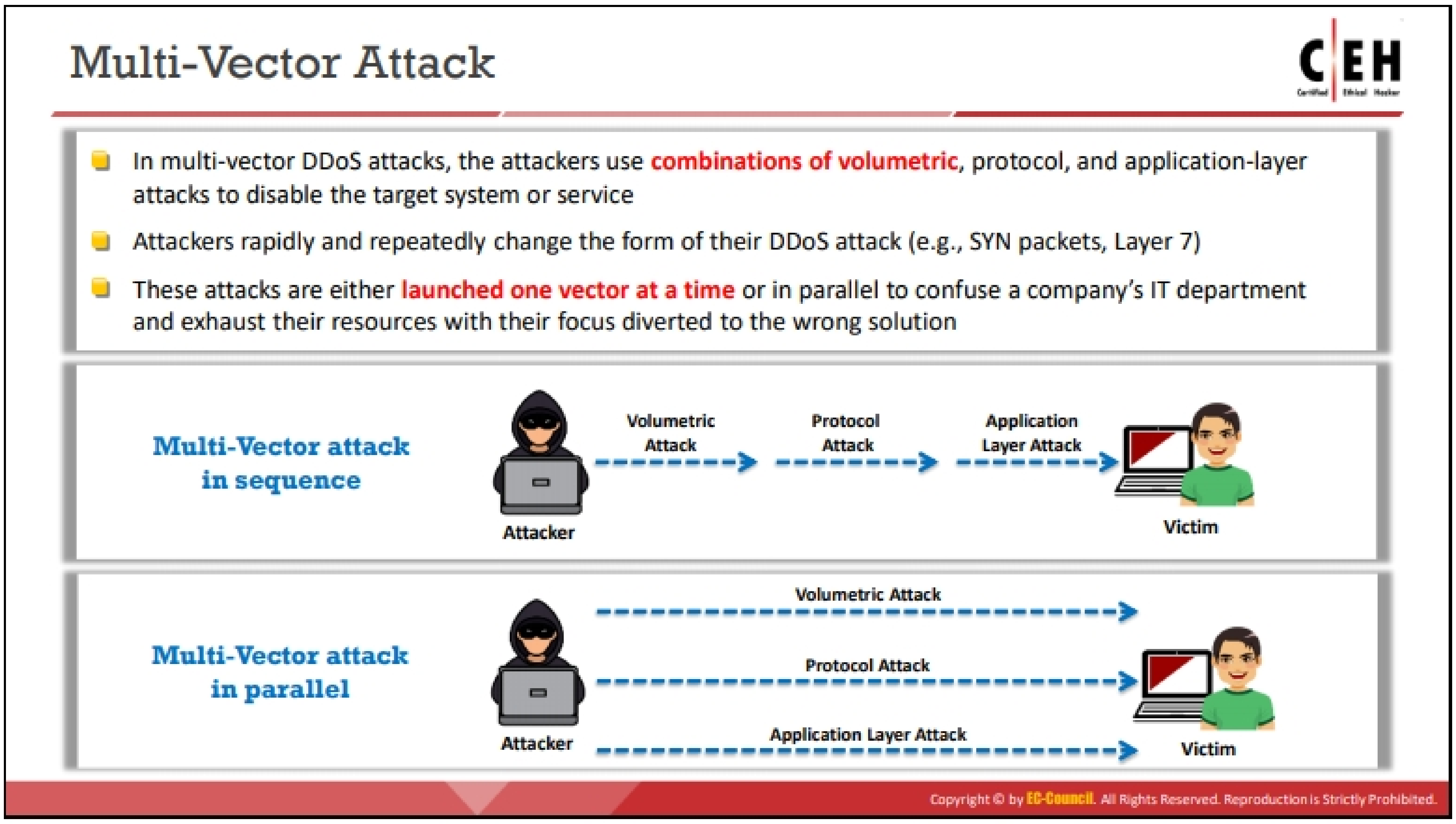
1. **CharGEN**
2. **SNMPv2**
3. **QOTD**
4. **RPC**
5. **SSDP**
6. **CLDAP**
7. **TFTP**
8. **NetBIOS**
9. **NTP**
10. **Quake Network Protocol**
11. **Steam Protocol**
12. **VoIP**

UDP Application Layer Flood Attack

Though UDP flood attacks are known for their volumetric attack nature, some application layer protocols that rely on UDP can be employed by attackers to perform flood attacks on target networks.

The following are examples for UDP-based application layer protocols that attackers can employ for flooding target networks:

- Character Generator Protocol (CHARGEN)
- Simple Network Management Protocol Version 2 (SNMPv2)
- Quote of the Day (QOTD)
- Remote procedure call (RPC)
- SSDP
- Connection-less Lightweight Directory Access Protocol (CLDAP)
- Trivial File Transfer Protocol (TFTP)
- Network Basic Input/Output System (NetBIOS)
- NTP
- Quake Network Protocol
- Steam Protocol
- Voice over Internet Protocol (VoIP)



Multi-Vector Attack

- In multi-vector DDoS attacks, the attackers use **combinations of volumetric**, protocol, and application-layer attacks to disable the target system or service
- Attackers rapidly and repeatedly change the form of their DDoS attack (e.g., SYN packets, Layer 7)
- These attacks are either **launched one vector at a time** or in parallel to confuse a company's IT department and exhaust their resources with their focus diverted to the wrong solution

Multi-Vector attack in sequence

Attacker → Volumetric Attack → Protocol Attack → Application Layer Attack → Victim

Multi-Vector attack in parallel

Attacker → Volumetric Attack → Victim
Attacker → Protocol Attack → Victim
Attacker → Application Layer Attack → Victim

Multi-Vector Attack

In multi-vector DDoS attacks, the attacker uses combinations of volumetric, protocol, and application layer attacks to take down the target system or service. The attacker quickly changes from one form of DDoS attack (e.g., SYN packets) to another (layer 7). These attacks are either launched through one vector at a time or through multiple vectors in parallel to confuse a company's IT department, making them spend all their resources and maliciously diverting their focus.

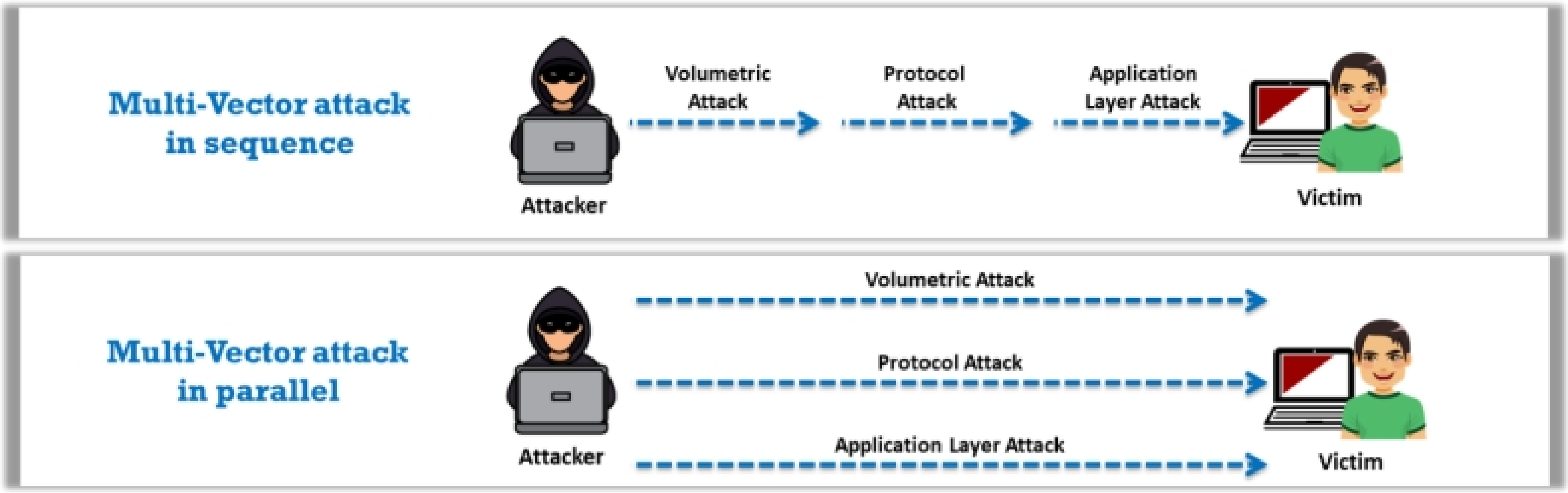
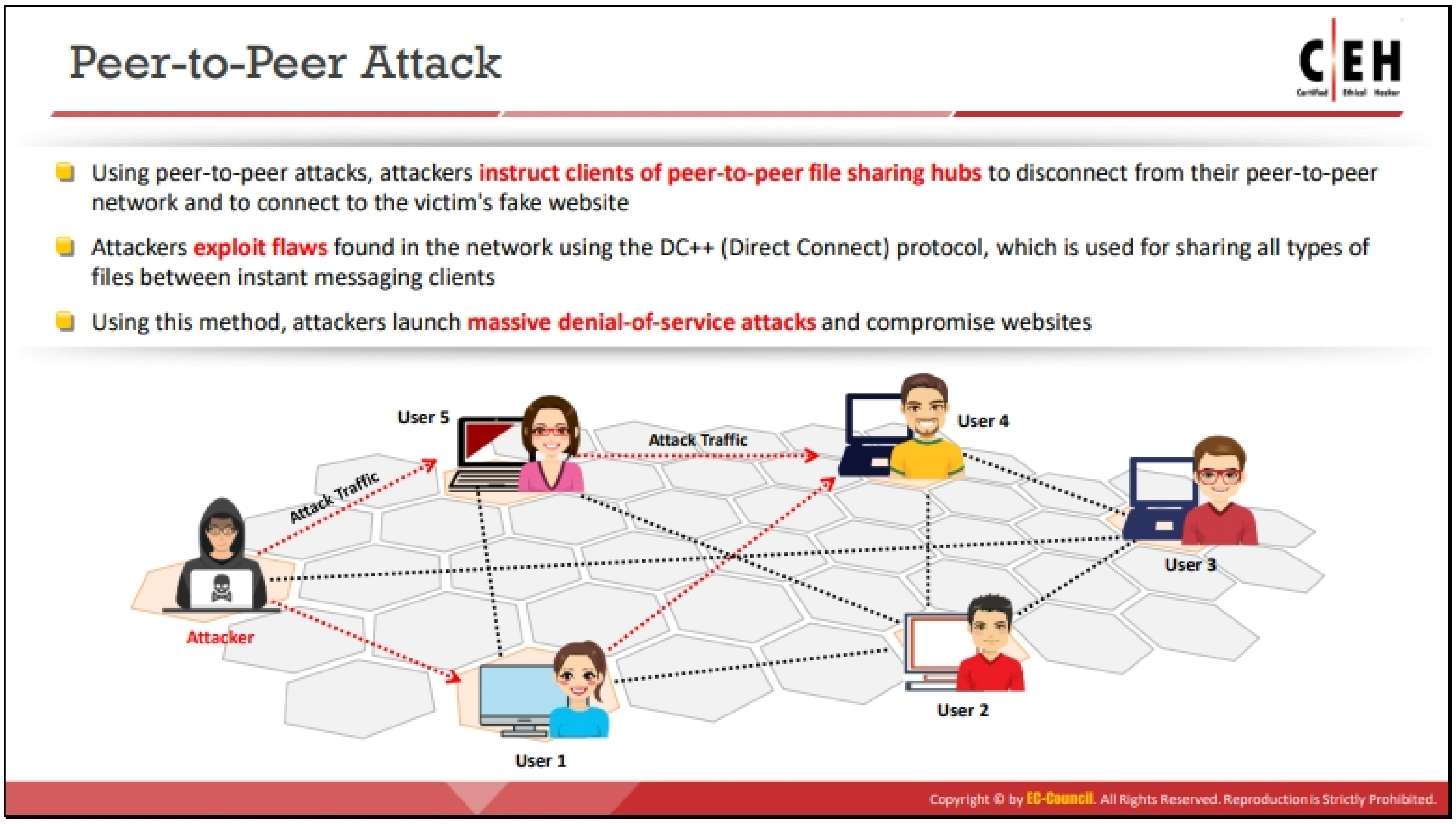

Multi-Vector attack in sequence
Volumetric Attack
Protocol Attack
Application Layer Attack
Attacker
Victim
Multi-Vector attack in parallel
Volumetric Attack
Protocol Attack
Application Layer Attack
Attacker
Victim

Figure 10.12: Multi-vector attack



Peer-to-Peer Attack

- Using peer-to-peer attacks, attackers **instruct clients of peer-to-peer file sharing hubs** to disconnect from their peer-to-peer network and to connect to the victim's fake website
- Attackers **exploit flaws** found in the network using the DC++ (Direct Connect) protocol, which is used for sharing all types of files between instant messaging clients
- Using this method, attackers launch **massive denial-of-service attacks** and compromise websites

Peer-to-Peer Attack

A peer-to-peer attack is a form of DDoS attack in which the attacker exploits a number of bugs in peer-to-peer servers to initiate a DDoS attack. Attackers exploit flaws found in networks that use the Direct Connect (DC++) protocol, which allows the exchange of files between instant-messaging clients. This kind of attack does not use botnets. Unlike a botnet-based attack, a peer-to-peer attack eliminates the need for attackers to communicate with the clients they subvert. Here, the attacker instructs clients of large peer-to-peer file sharing hubs to disconnect from their peer-to-peer network and instead connect to the victim's website. Consequently, several thousand computers may aggressively attempt to connect to a target website, causing a drop in the performance of the target website. It is easy to identify peer-to-peer attacks based on signatures. By using this method, attackers launch massive DoS attacks to compromise websites.

Peer-to-peer DDoS attacks can be minimized by specifying ports for peer-to-peer communication. For example, specifying port 80 to disallow peer-to-peer communication minimizes the possibility of attacks on websites.

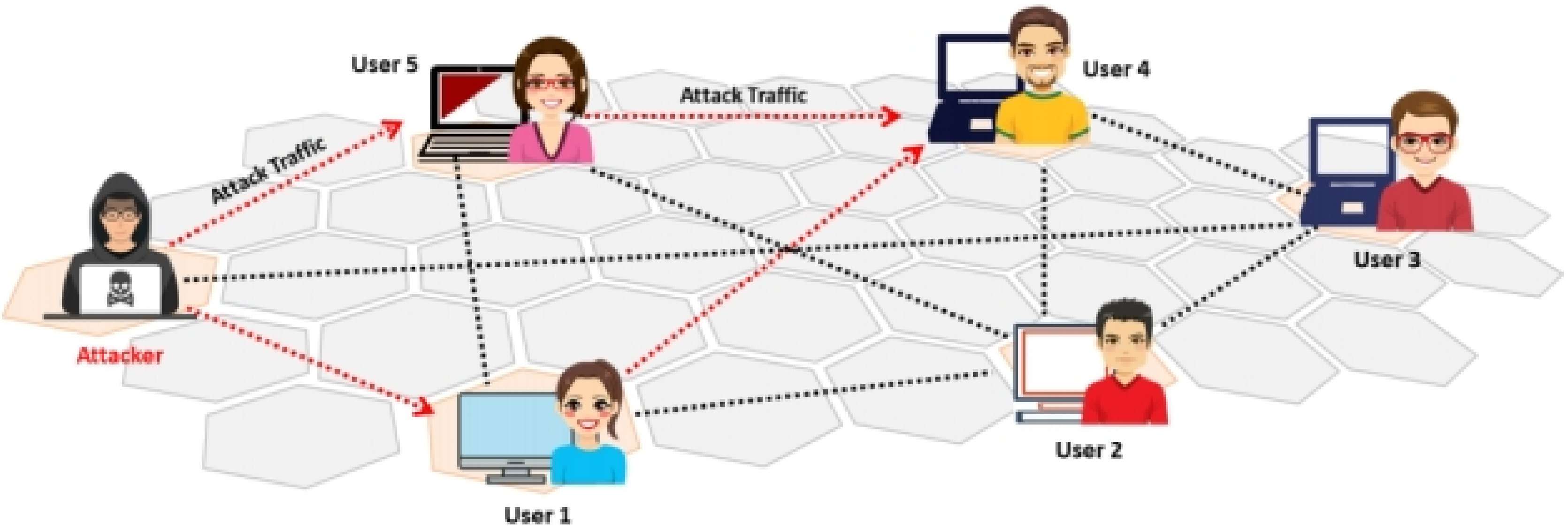
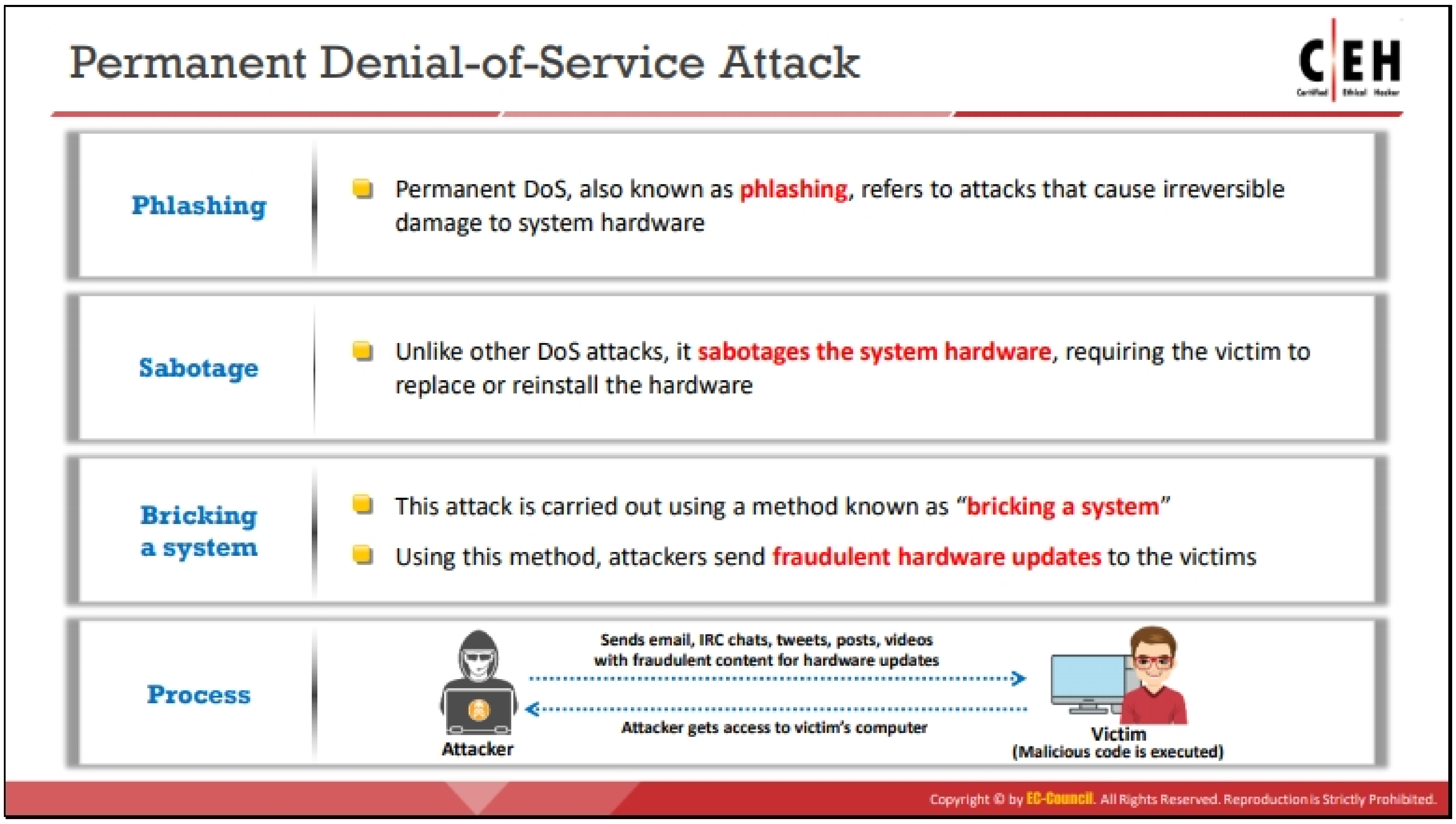


Figure 10.13: Peer-to-peer attack



Permanent Denial-of-Service Attack

Phlashing

- Permanent DoS, also known as **phlashing**, refers to attacks that cause irreversible damage to system hardware

Sabotage

- Unlike other DoS attacks, it **sabotages the system hardware**, requiring the victim to replace or reinstall the hardware

Bricking a system

- This attack is carried out using a method known as "**bricking a system**"
- Using this method, attackers send **fraudulent hardware updates** to the victims

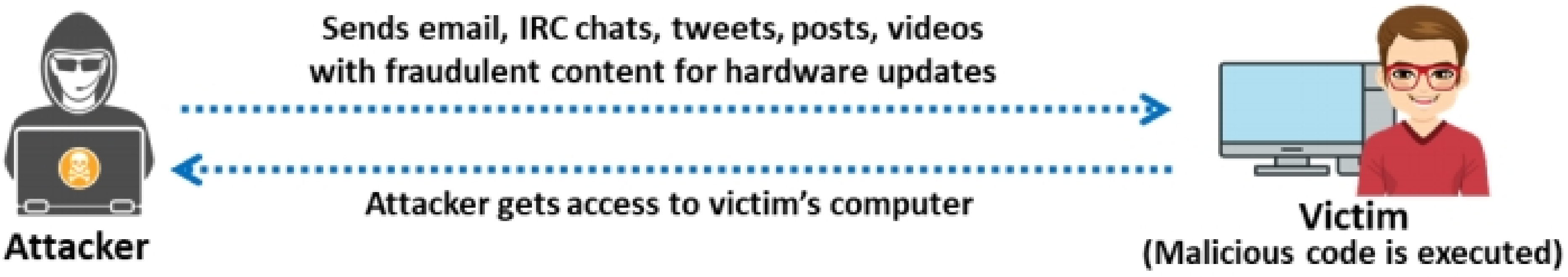
Process

Sends email, IRC chats, tweets, posts, videos with fraudulent content for hardware updates
Attacker gets access to victim's computer
Attacker
Victim
(Malicious code is executed)

Permanent Denial-of-Service Attack

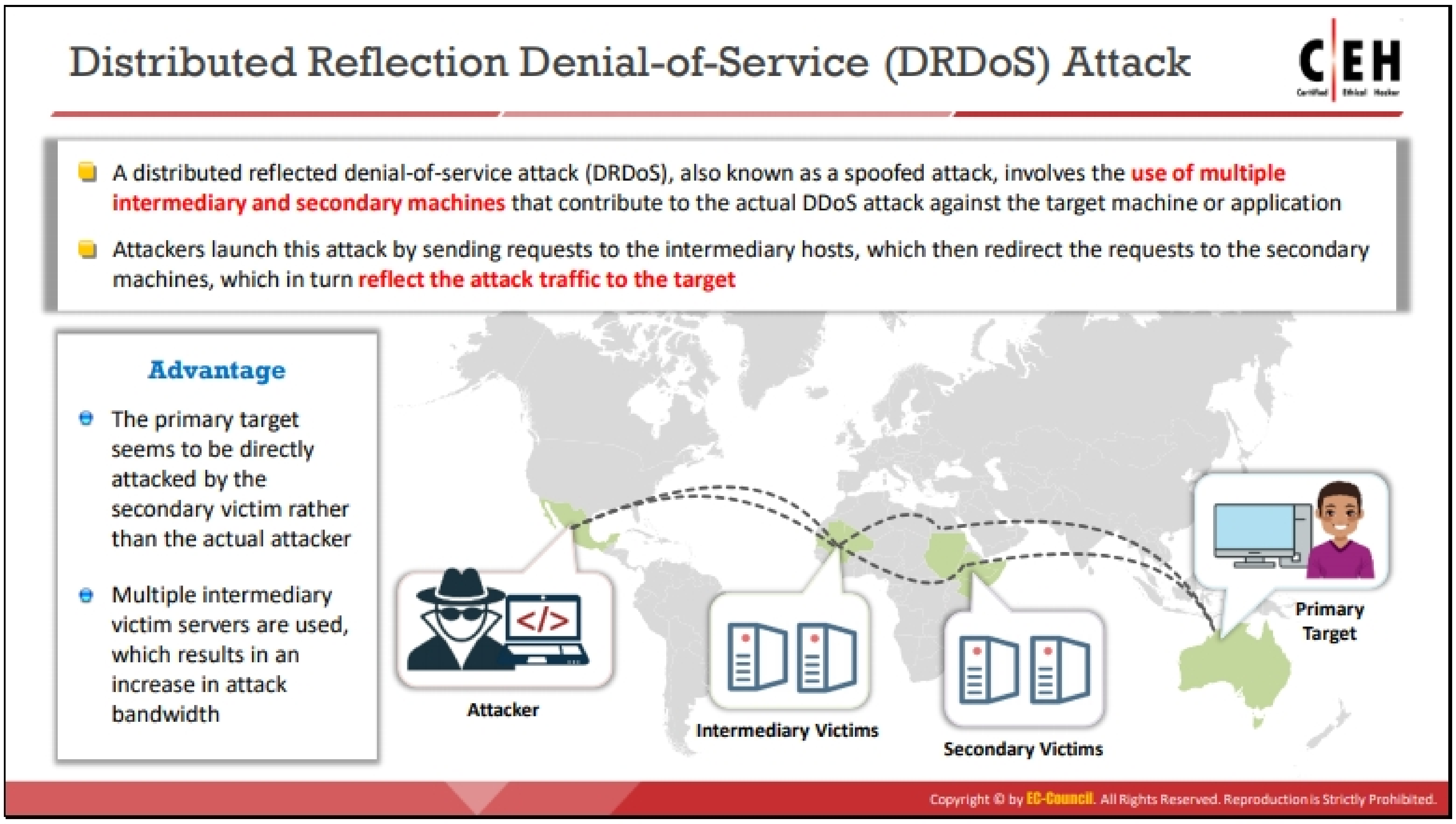
Permanent DoS (PDoS) attacks, also known as phlashing, purely target hardware and cause irreversible damage to the hardware. Unlike other types of DoS attacks, it sabotages the system hardware, requiring the victim to replace or reinstall the hardware. The PDoS attack exploits security flaws in a device to allow remote administration on the management interfaces of the victim's hardware, such as printers, routers, and other networking devices.

This type of attack is quicker and more destructive than conventional DoS attacks. It works with a limited amount of resources, unlike a DDoS attack, in which attackers unleash a set of zombies onto a target. Attackers perform PDoS attacks by using a method known as the "bricking" of a system. In this method, the attacker sends emails, IRC chats, tweets, or videos with fraudulent content for hardware updates to the victim. The hardware updates are modified and corrupted with vulnerabilities or defective firmware. When the victim clicks on a link or pop-up window referring to the fraudulent hardware update, the victim installs it in their system. Consequently, the attacker attains complete control over the victim's system.



Sends email, IRC chats, tweets, posts, videos with fraudulent content for hardware updates
Attacker gets access to victim's computer
Attacker
Victim
(Malicious code is executed)

Figure 10.14: Permanent DoS attack



Distributed Reflection Denial-of-Service (DRDoS) Attack

- A distributed reflected denial-of-service attack (DRDoS), also known as a spoofed attack, involves the **use of multiple intermediary and secondary machines** that contribute to the actual DDoS attack against the target machine or application
- Attackers launch this attack by sending requests to the intermediary hosts, which then redirect the requests to the secondary machines, which in turn **reflect the attack traffic to the target**

Advantage

- The primary target seems to be directly attacked by the secondary victim rather than the actual attacker
- Multiple intermediary victim servers are used, which results in an increase in attack bandwidth

Distributed Reflection Denial-of-Service (DRDoS) Attack

A distributed reflection DoS (DRDoS) attack, also known as a “spoofed” attack, involves the use of multiple intermediary and secondary machines that contribute to a DDoS attack against a target machine or application. A DRDoS attack exploits the TCP three-way handshake vulnerability.

This attack involves an attacker machine, intermediary victims (zombies), secondary victims (reflectors), and a target machine. The attacker launches this attack by sending requests to the intermediary hosts, which in turn reflect the attack traffic to the target.

The process of a DRDoS attack is as follows. First, the attacker commands the intermediary victims (zombies) to send a stream of packets (TCP SYN) with the primary target’s IP address as the source IP address to other non-compromised machines (secondary victims or reflectors) in order to exhort them to establish a connection with the primary target. Consequently, the reflectors send a huge volume of traffic (SYN/ACK) to the primary target to establish a new connection with it because they believe the host requested it. The primary target discards the SYN/ACK packets received from the reflectors because they did not send the SYN packet. Meanwhile, the reflectors wait for the ACK response from the primary target. Assuming that the packet was lost, the reflector machines resend SYN/ACK packets to the primary target to establish the connection, until a time-out occurs. In this manner, the target machine is flooded with a heavy volume of traffic from the reflector machines. The combined bandwidth of these reflector machines overwhelms the target machine.

A DRDoS attack is an intelligent attack because it is very difficult or even impossible to trace the attacker. Instead of the actual attacker, the secondary victims (reflectors) seem to attack the primary target directly. This attack is more effective than a typical DDoS attack because multiple intermediary and secondary victims generate huge attack bandwidth.

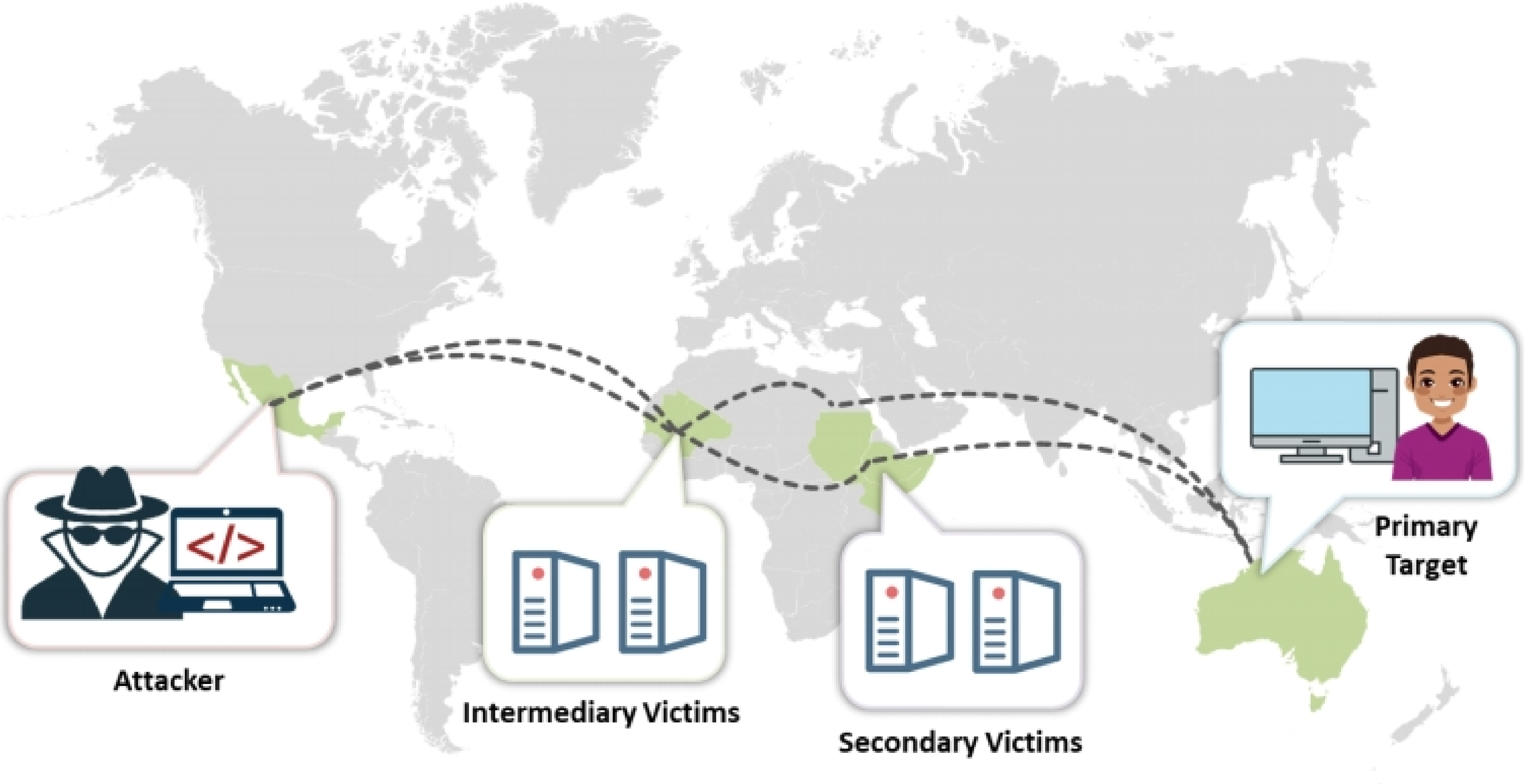
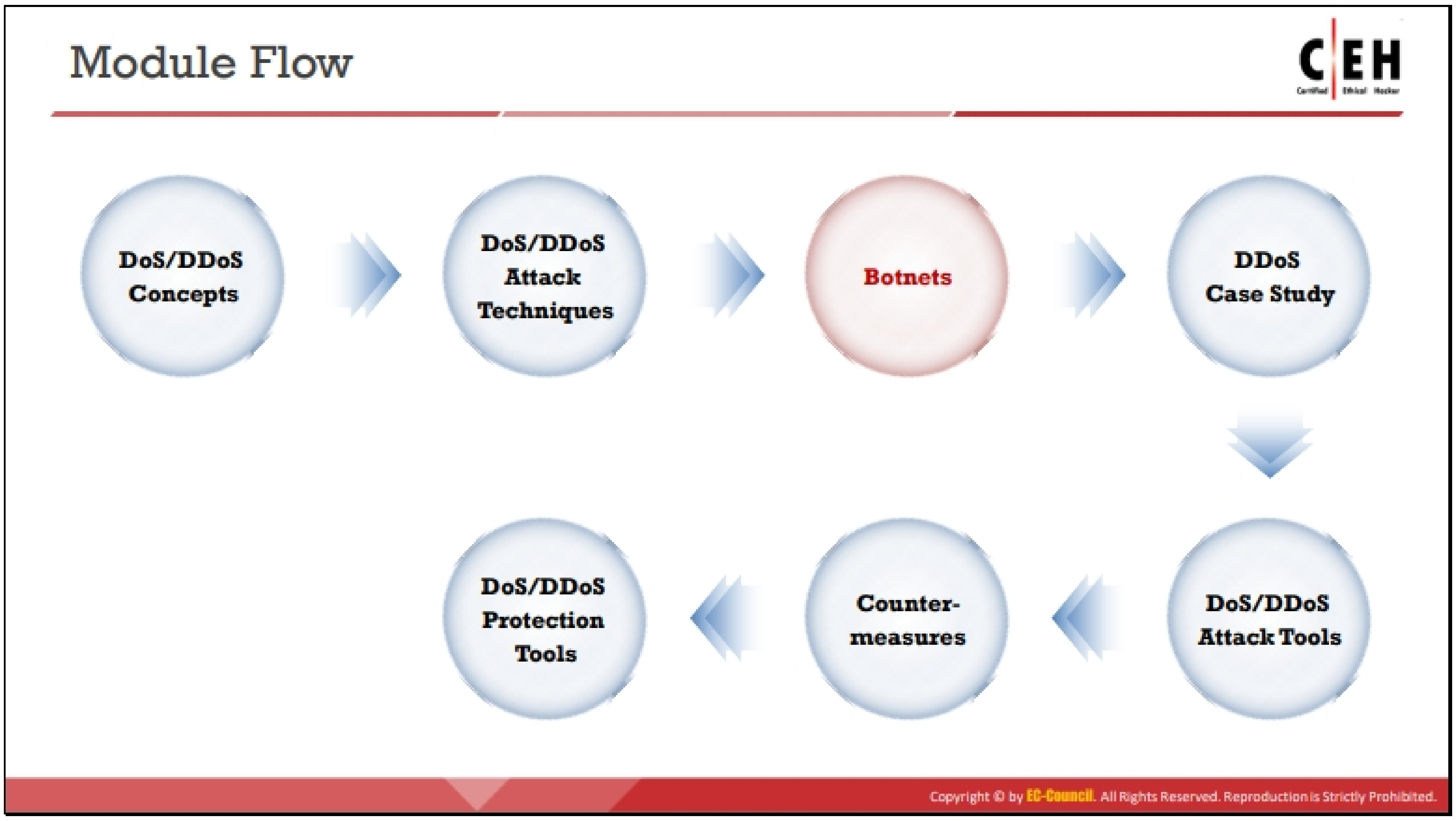

Attacker
Intermediary Victims
Secondary Victims
Primary Target

Figure 10.15: Distributed reflection DoS (DRDoS) attack

- **Countermeasures**
  - Turn off the Character Generator Protocol (CHARGEN) service to stop this attack method
  - Download the latest updates and patches for servers



Module Flow

DoS/DDoS Concepts

DoS/DDoS Attack Techniques

Botnets

DDoS Case Study

DoS/DDoS Attack Tools

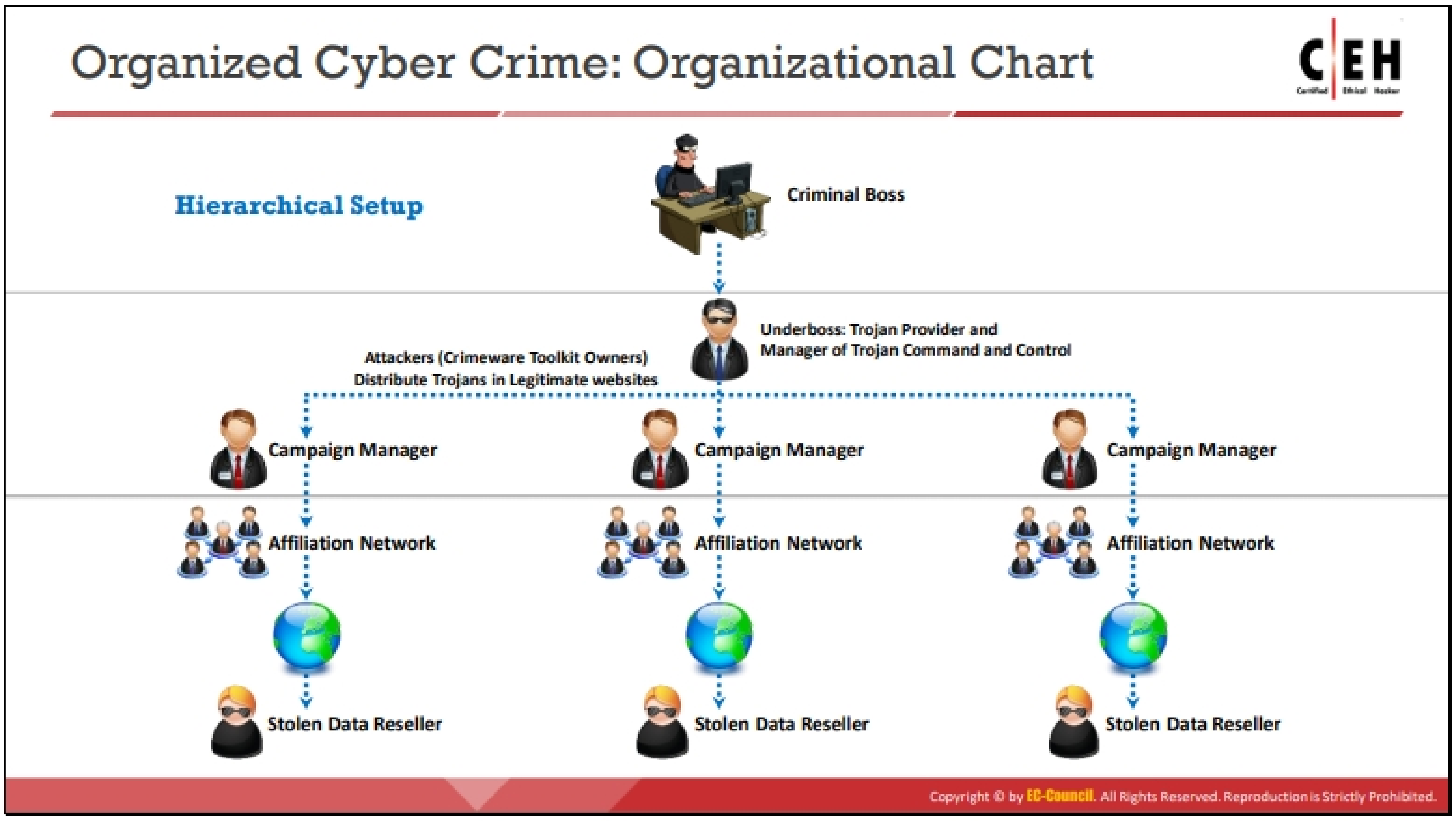
Counter-measures

DoS/DDoS Protection Tools

Botnets

The term “bot” is a contraction of “robot” and refers to software applications that run automated tasks over the Internet. Attackers use bots to infect a large number of computers that form a network, or “botnet,” allowing them to launch DDoS attacks, generate spam, spread viruses, and commit other types of crime.

This section deals with organized cyber-crime syndicates, organizational charts, botnets, and botnet propagation techniques; botnet ecosystems; scanning methods for finding vulnerable machines; and the propagation of malicious code.



Organized Cyber Crime: Organizational Chart

Hierarchical Setup

Criminal Boss

Underboss: Trojan Provider and Manager of Trojan Command and Control

Attackers (Crimeware Toolkit Owners) Distribute Trojans in Legitimate websites

Campaign Manager

Campaign Manager

Campaign Manager

Affiliation Network

Affiliation Network

Affiliation Network

Stolen Data Reseller

Stolen Data Reseller

Stolen Data Reseller

Organized Cyber Crime: Organizational Chart

Organized Crime Syndicates

While cyber criminals worked independently in the past, they now tend to operate in organized groups. They are increasingly associated with organized crime syndicates and take advantage of the sophisticated techniques of these syndicates to engage in illegal activity, usually for monetary benefit. There are organized groups of cyber criminals who work in a hierarchical set up with a predefined revenue-sharing model, which is a kind of major corporation that offers criminal services. Organized groups create and rent botnets and offer various services ranging from the development of malware and hacking of bank accounts to the deployment of massive DoS attacks against any target for a price.

For example, an organized crime syndicate might perform a DDoS attack against a bank to divert the attention of the bank's security team while they clean out bank accounts with stolen account credentials. The growing involvement of organized criminal syndicates in politically motivated cyber warfare and hacktivism is a matter of concern for national security agencies.

Cybercrime features a complicated range of players, and cyber criminals are paid according to the task they perform or the position they hold. The head of the cybercrime organization (i.e., the boss) acts as a business entrepreneur. The boss does not commit any crimes directly. Immediately below the boss in the organizational hierarchy is the "underboss," who sets up a C&C server and crimeware toolkit database to manage the implementation of attacks and provide Trojans. Below the underboss are various "campaign managers" with their own affiliation networks for implementing attacks and stealing data. Finally, resellers sell the stolen data.

Hierarchical Setup

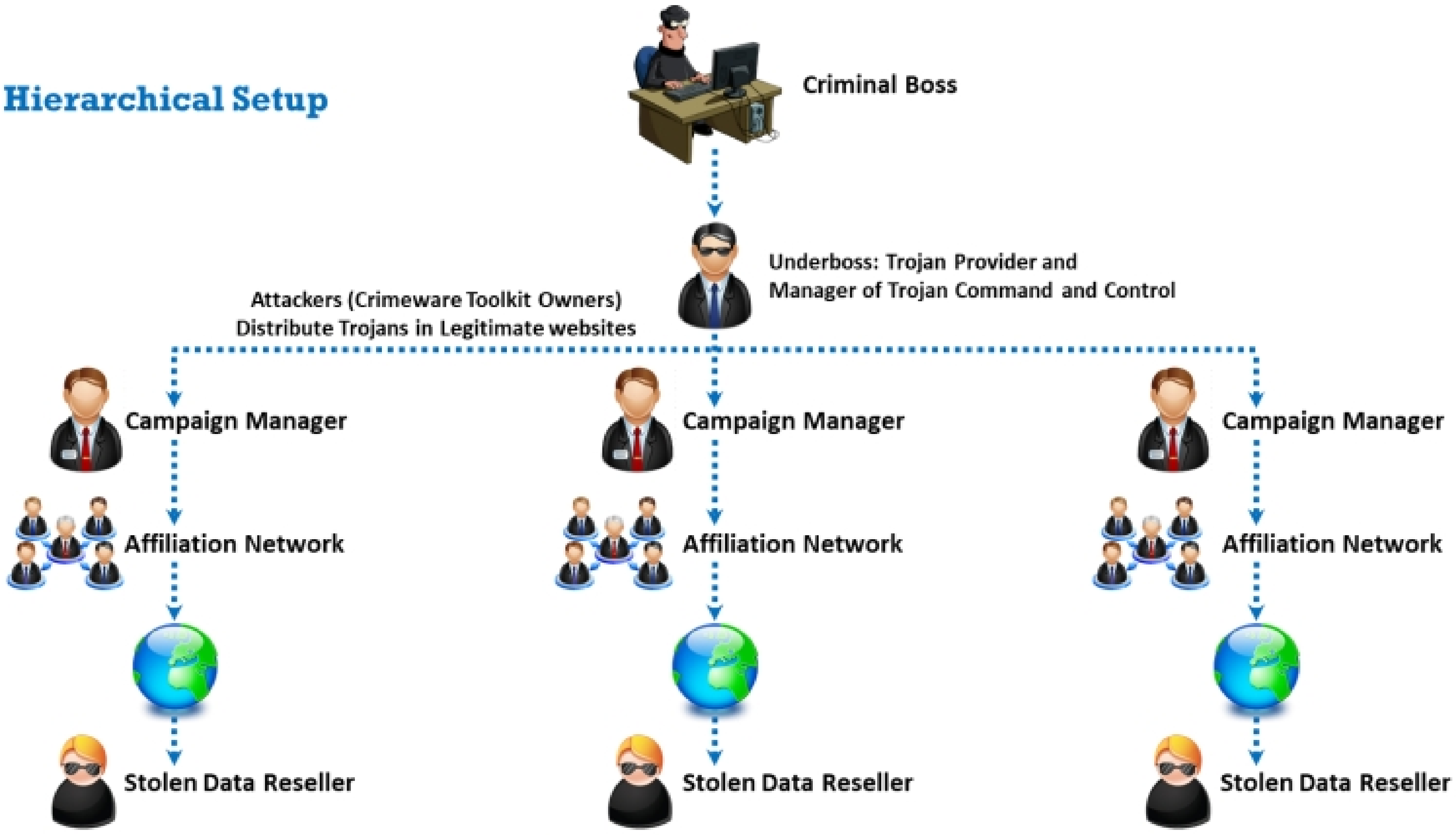
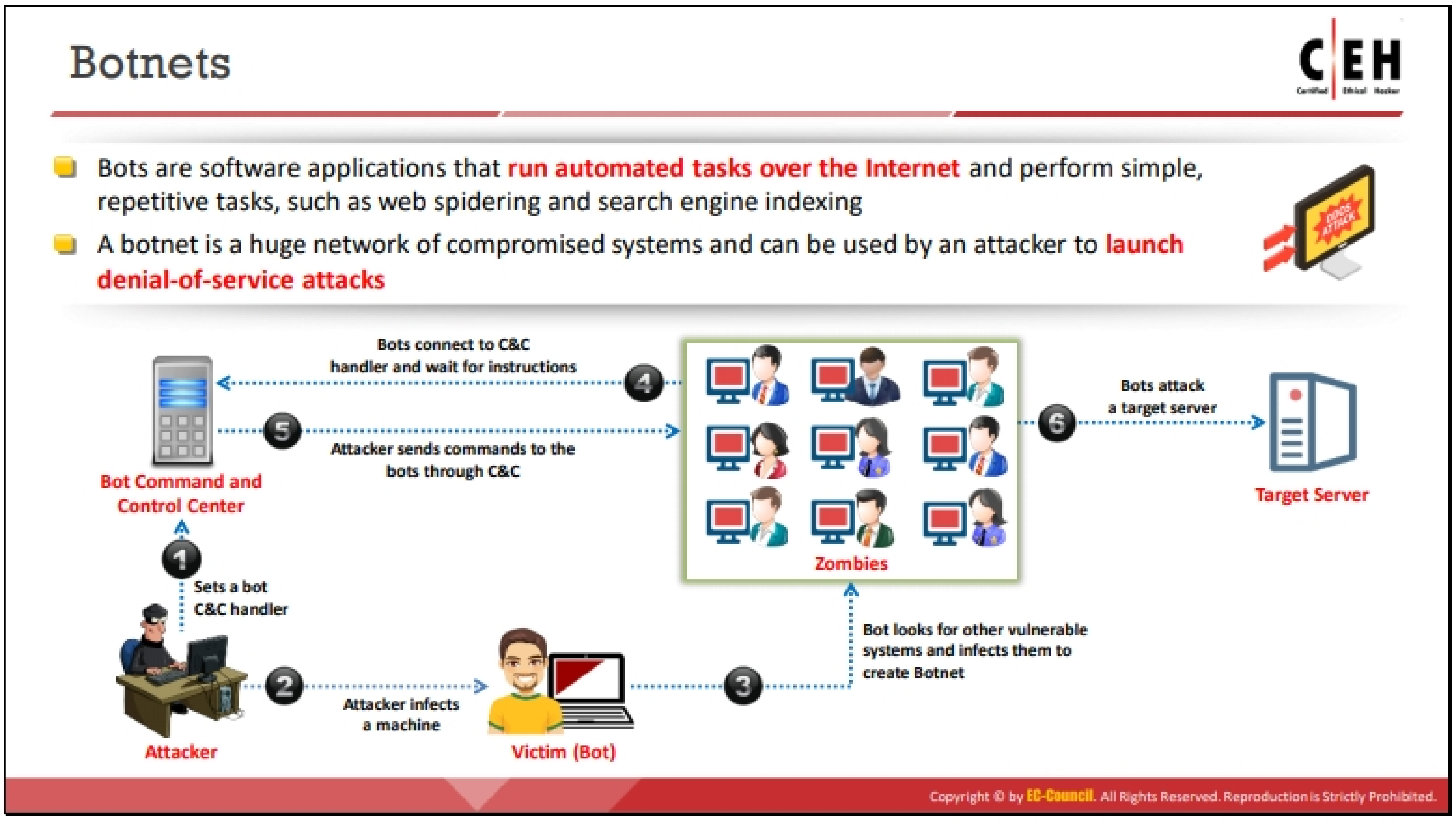

Criminal Boss
Underboss: Trojan Provider and Manager of Trojan Command and Control
Attackers (Crimeware Toolkit Owners) Distribute Trojans in Legitimate websites
Campaign Manager
Campaign Manager
Campaign Manager
Affiliation Network
Affiliation Network
Affiliation Network
Stolen Data Reseller
Stolen Data Reseller
Stolen Data Reseller

Figure 10.16: Hierarchical setup of a cybercrime organization



Botnets

- Bots are software applications that **run automated tasks over the Internet** and perform simple, repetitive tasks, such as web spidering and search engine indexing
- A botnet is a huge network of compromised systems and can be used by an attacker to **launch denial-of-service attacks**

Bot Command and Control Center

Zombies

Target Server

Attacker

Victim (Bot)

1 Sets a bot C&C handler

2 Attacker infects a machine

3 Bot looks for other vulnerable systems and infects them to create Botnet

4 Bots connect to C&C handler and wait for instructions

5 Attacker sends commands to the bots through C&C

6 Bots attack a target server

Botnets

Bots are used for benign data collection or data mining activities, such as "web spidering," as well as to coordinate DoS attacks. The main purpose of a bot is to collect data. There are different types of bots, such as Internet bots, IRC bots, and chatter bots. Examples for IRC bots are Supybot, Sopel, EnergyMech, and Eggdrop.

A botnet (a contraction of "roBOT NETwork") is a group of computers "infected" by bots; however, botnets can be used for both positive and negative purposes. As a hacking tool, a botnet is composed of a huge network of compromised systems. A relatively small botnet of 1,000 bots has a combined bandwidth larger than the bandwidth of most corporate systems.

The advent of botnets led to an enormous increase in cybercrime. Botnets form the core of the cybercriminal activity center that links and unites various parts of the cybercriminal world. Cybercriminal service suppliers are a part of a cybercrime network. They offer services such as malicious code development, bulletproof hosting, the creation of browser exploits, and encryption and packing.

Malicious code is the primary tool used by criminal organizations to commit cybercrimes. Botnet owners order both bots and other malicious programs such as Trojans, viruses, worms, keyloggers, and specially crafted applications to attack remote computers via networks. Developers offer malware services on public sites or closed Internet resources.

Botnets are agents that an intruder can send to a server system to perform an illegal activity. Botnets run hidden programs that allow the identification of system vulnerabilities. Attackers can use botnets to perform the tedious tasks involved in probing a system for known vulnerabilities.

Attackers can use botnets to perform the following:

- **DDoS attacks**: Botnets can generate DDoS attacks, which consume the bandwidth of the victim's computers. Botnets can also overload a system, wasting valuable host system resources and destroying network connectivity.
- **Spamming**: Attackers use a SOCKS proxy for spamming. They harvest email addresses from web pages or other sources.
- **Sniffing traffic**: A packet sniffer observes the data traffic entering a compromised machine. It allows an attacker to collect sensitive information such as credit card numbers and passwords. The sniffer also allows an attacker to steal information from one botnet and use it against another botnet. In other words, botnets can rob one another.
- **Keylogging**: Keylogging is a method of recording the keys typed on a keyboard, and it provides sensitive information such as system passwords. Attackers use keylogging to harvest account login information for services such as PayPal.
- **Spreading new malware**: Botnets can be used to spread new bots.
- **Installing advertisement add-ons**: Botnets can be used to perpetrate a "click fraud" by automating clicks.
- **Google AdSense abuse**: Some companies permit showing Google AdSense ads on their websites for economic benefits. Botnets allow an intruder to automate clicks on an ad, producing a percentage increase in the click queue.
- **Attacks on IRC chat networks**: Also called clone attacks, these attacks are similar to a DDoS attack. A master agent instructs each bot to link to thousands of clones within an IRC network, which can flood the network.
- **Manipulating online polls and games**: Every botnet has a unique address, enabling it to manipulate online polls and games.
- **Mass identity theft**: Botnets can send a large number of emails while impersonating a reputable organization such as eBay. This technique allows attackers to steal information for identity theft.

The below figure illustrates how an attacker launches a botnet-based DoS attack on a target server. The attacker sets up a bot C&C center, following which they infect a machine (bot) and compromises it. Later, they use this bot to infect and compromise other vulnerable systems available in the network, resulting in a botnet. The bots (also known as zombies) connect to the C&C center and awaits instructions. Subsequently, the attacker sends malicious commands to the bots through the C&C center. Finally, as per the attacker's instructions, the bots launch a DoS attack on a target server, making its services unavailable to legitimate users in the network.

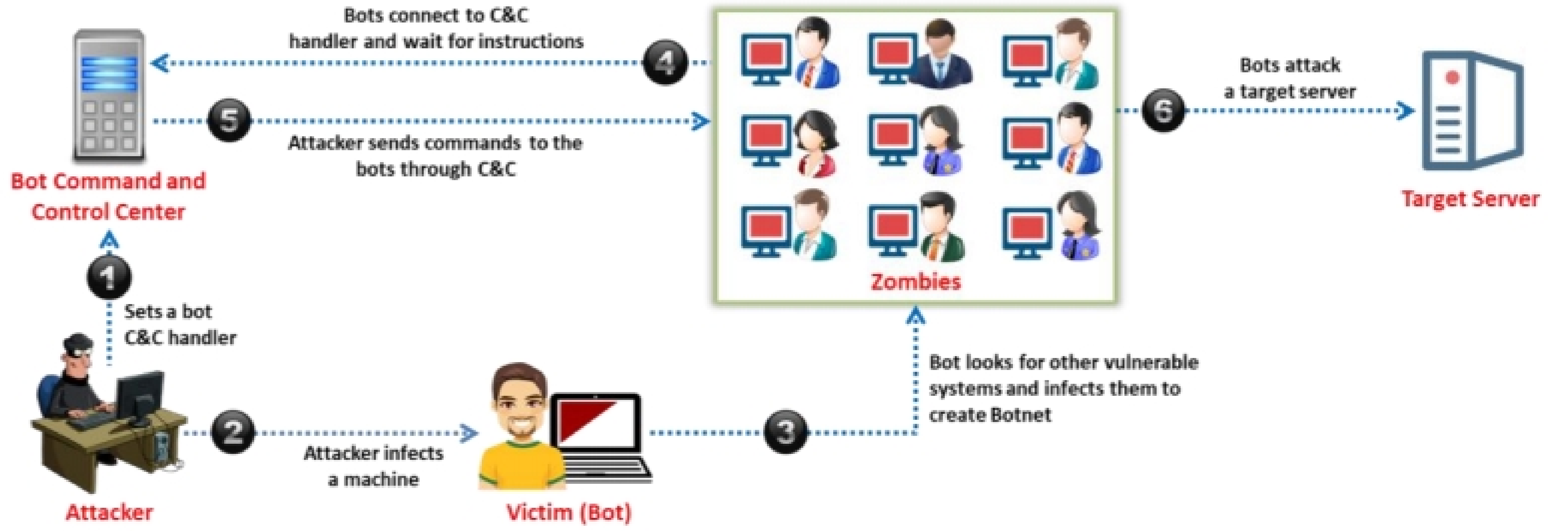
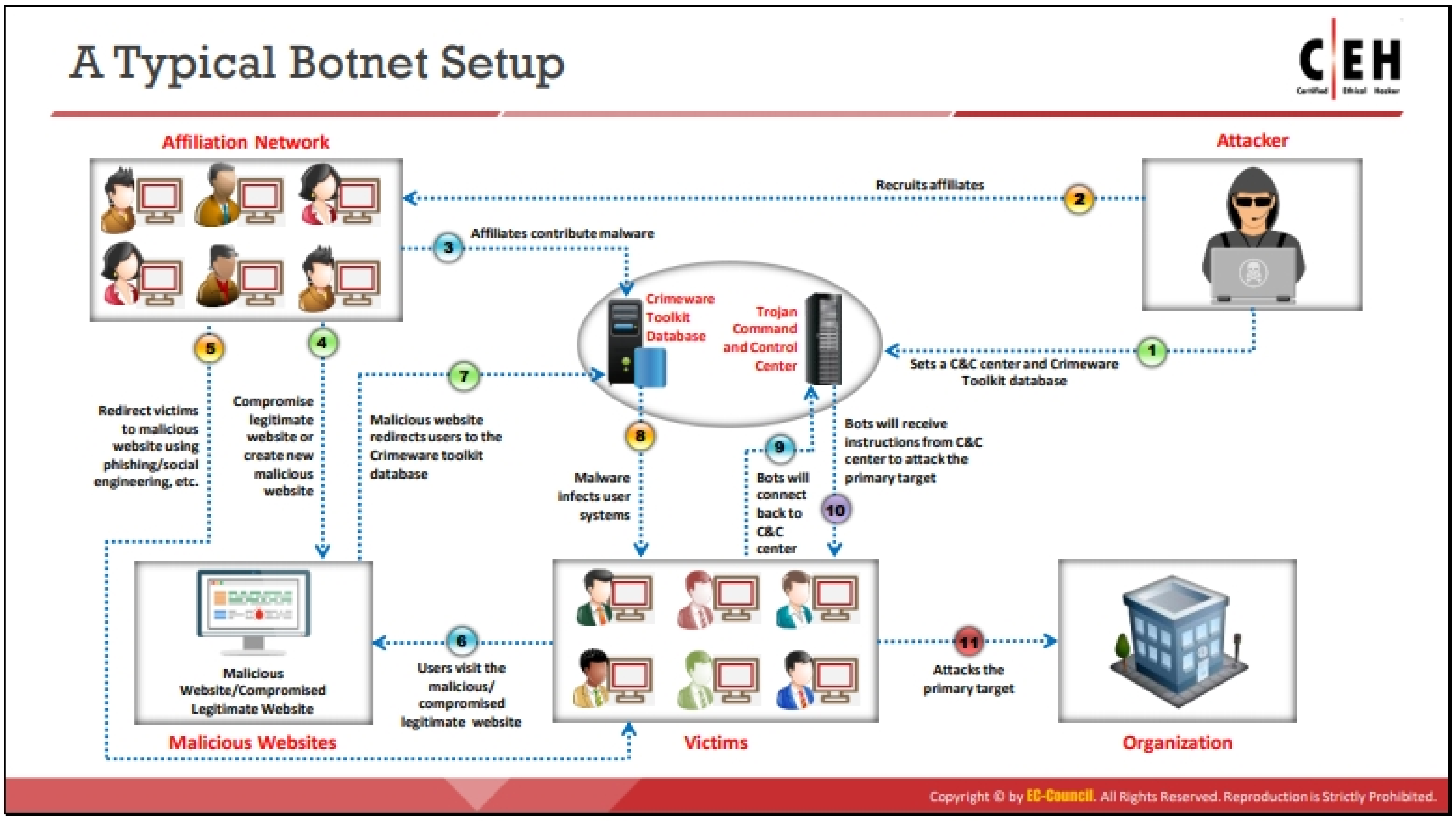

Bots connect to C&C handler and wait for instructions
4
5
Attacker sends commands to the bots through C&C
Bot Command and Control Center
Zombies
6
Bots attack a target server
Target Server
1
Sets a bot C&C handler
2
Attacker infects a machine
Attacker
Victim (Bot)
3
Bot looks for other vulnerable systems and infects them to create Botnet

Figure 10.17: Botnet-based DDoS attack



A Typical Botnet Setup

Affiliation Network

Attacker

Recruits affiliates 2

3 Affiliates contribute malware

Crimeware Toolkit Database

Trojan Command and Control Center

1 Sets a C&C center and Crimeware Toolkit database

5 Redirect victims to malicious website using phishing/social engineering, etc.

4 Compromise legitimate website or create new malicious website

7 Malicious website redirects users to the Crimeware toolkit database

8 Malware infects user systems

9 Bots will connect back to C&C center

10 Bots will receive instructions from C&C center to attack the primary target

Malicious Website/Compromised Legitimate Website

6 Users visit the malicious/ compromised legitimate website

11 Attacks the primary target

Malicious Websites

Victims

Organization

A Typical Botnet Setup

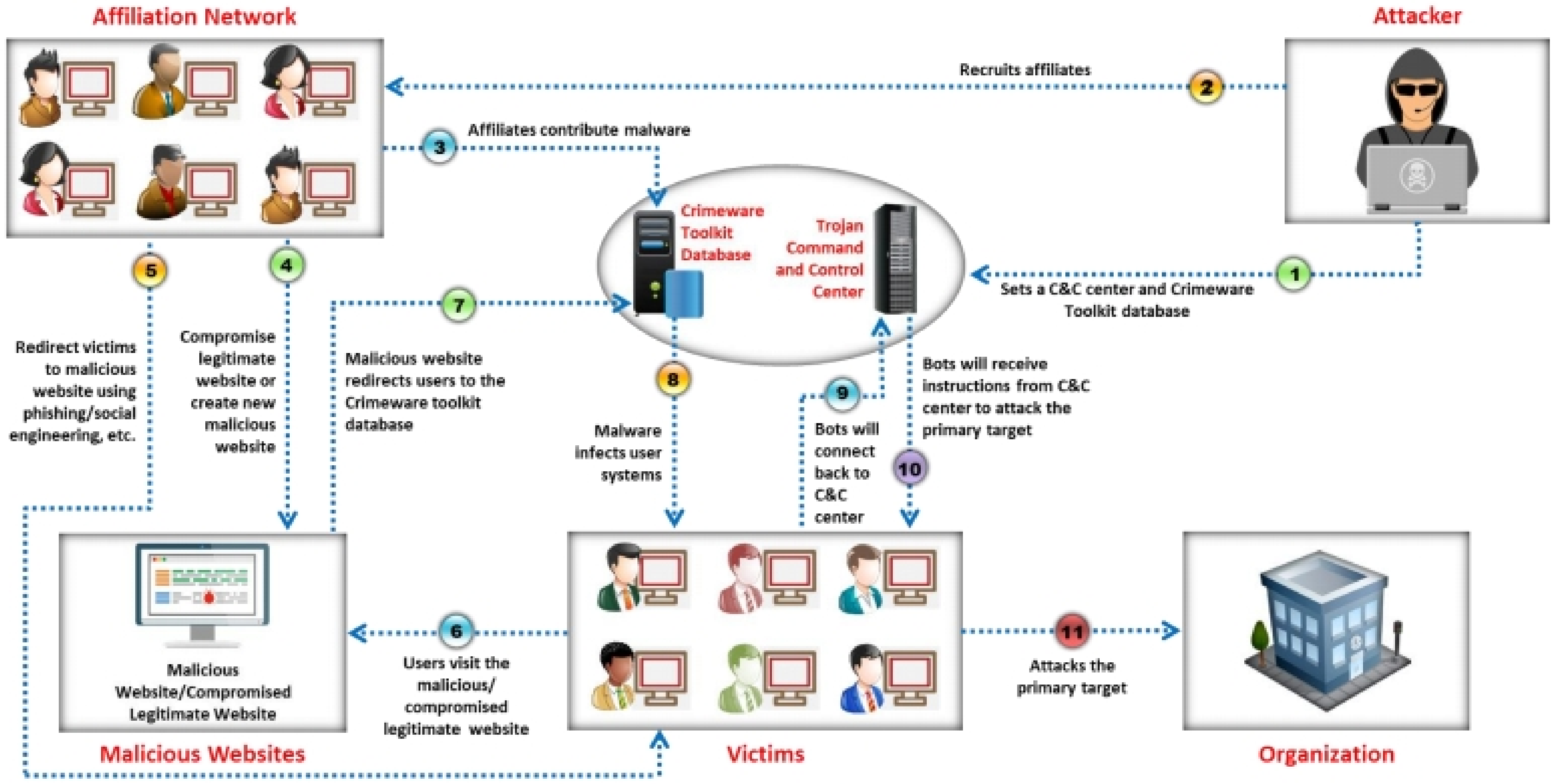
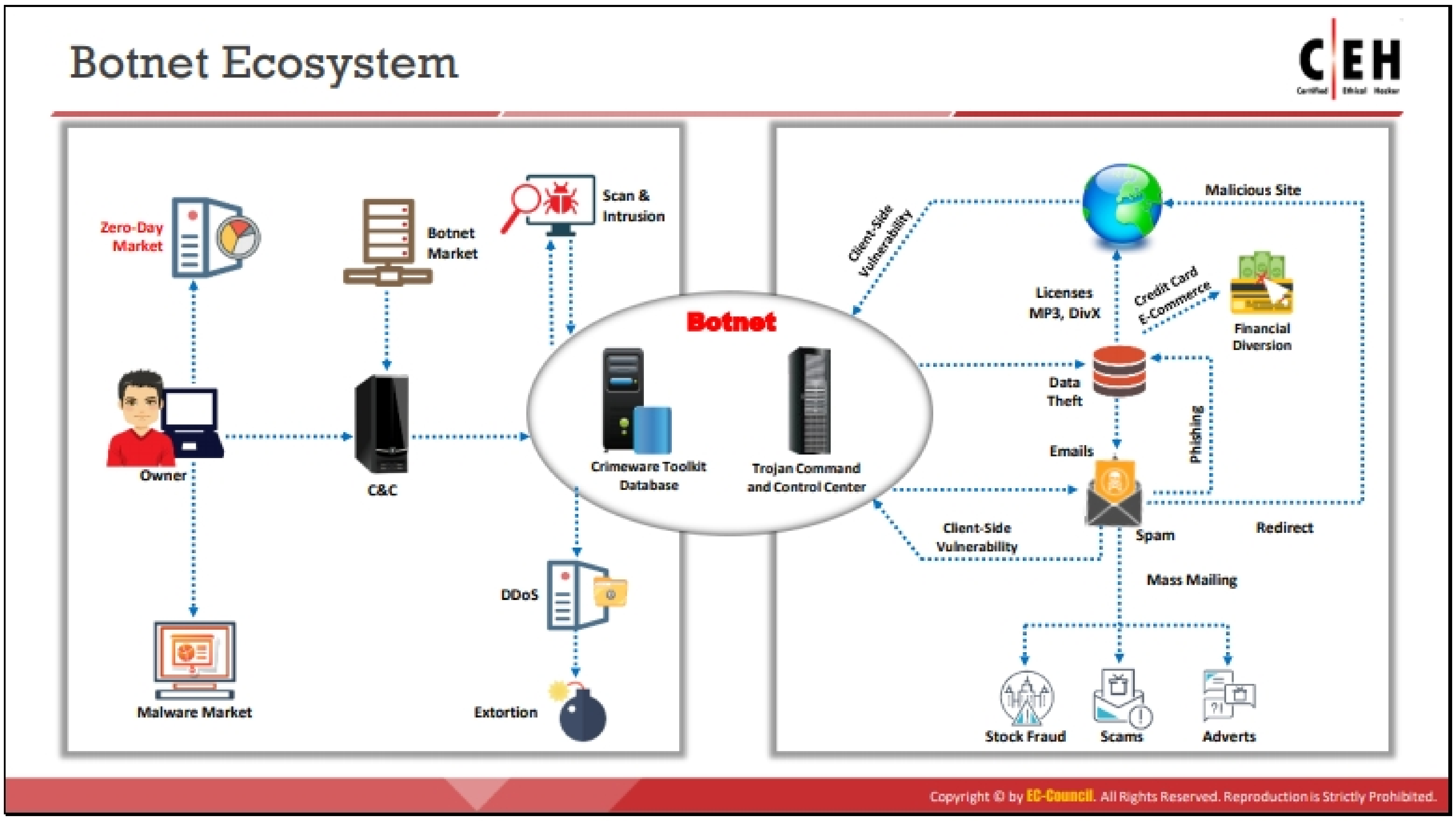

Affiliation Network
Attacker
Recruits affiliates
2
3
Affiliates contribute malware
Crimeware Toolkit Database
Trojan Command and Control Center
1
Sets a C&C center and Crimeware Toolkit database
5
4
7
Redirect victims to malicious website using phishing/social engineering, etc.
Compromise legitimate website or create new malicious website
Malicious website redirects users to the Crimeware toolkit database
8
Malware infects user systems
9
Bots will connect back to C&C center
10
Bots will receive instructions from C&C center to attack the primary target
Malicious Website/Compromised Legitimate Website
6
Users visit the malicious/ compromised legitimate website
11
Attacks the primary target
Malicious Websites
Victims
Organization

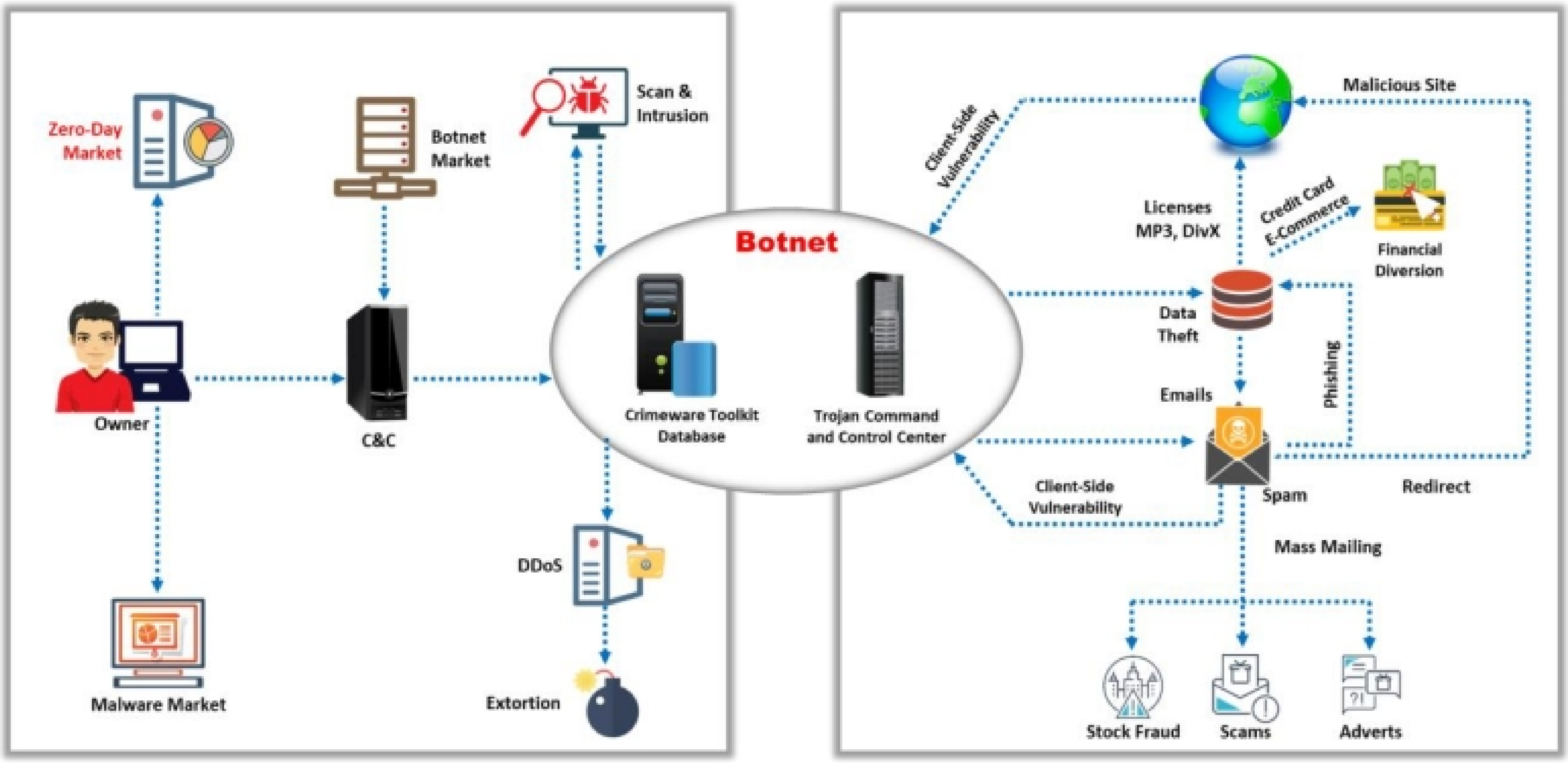
Figure 10.18: Typical botnet setup



Botnet Ecosystem

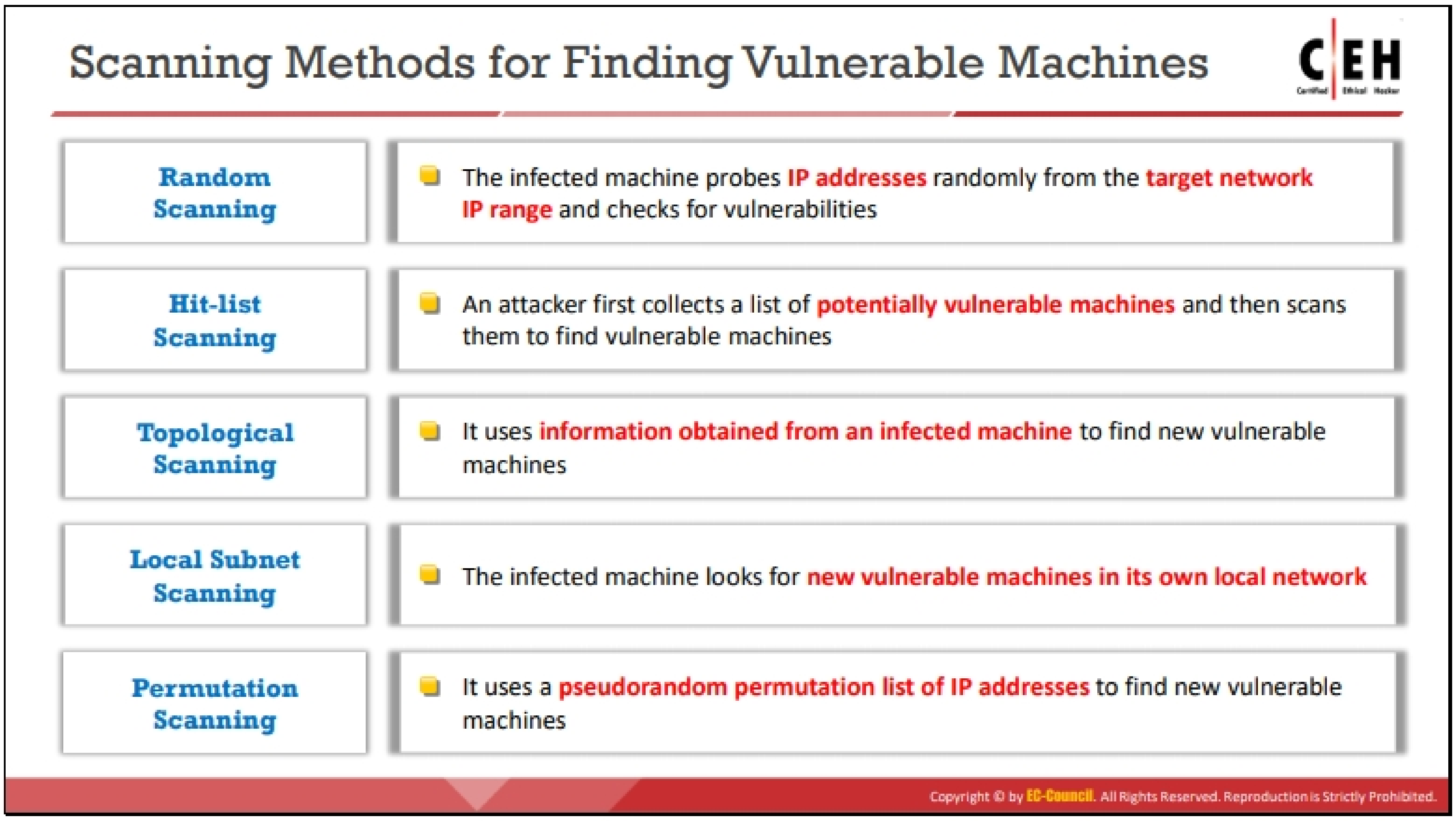
Zero-Day Market
Botnet Market
Scan & Intrusion
Owner
C&C
Botnet
Crimeware Toolkit Database
Trojan Command and Control Center
DDoS
Malware Market
Extortion
Client-Side Vulnerability
Malicious Site
Licenses MP3, DivX
Credit Card E-Commerce
Financial Diversion
Data Theft
Phishing
Emails
Spam
Redirect
Client-Side Vulnerability
Mass Mailing
Stock Fraud
Scams
Adverts

Botnet Ecosystem



Zero-Day Market
Botnet Market
Scan & Intrusion
Owner
C&C
Botnet
Crimeware Toolkit Database
Trojan Command and Control Center
DDoS
Malware Market
Extortion
Client-Side Vulnerability
Malicious Site
Licenses MP3, DivX
Credit Card E-Commerce
Financial Diversion
Data Theft
Phishing
Emails
Spam
Redirect
Client-Side Vulnerability
Mass Mailing
Stock Fraud
Scams
Adverts

Figure 10.19: Botnet ecosystem



Scanning Methods for Finding Vulnerable Machines

Random Scanning	The infected machine probes IP addresses randomly from the target network IP range and checks for vulnerabilities
Hit-list Scanning	An attacker first collects a list of potentially vulnerable machines and then scans them to find vulnerable machines
Topological Scanning	It uses information obtained from an infected machine to find new vulnerable machines
Local Subnet Scanning	The infected machine looks for new vulnerable machines in its own local network
Permutation Scanning	It uses a pseudorandom permutation list of IP addresses to find new vulnerable machines

Scanning Methods for Finding Vulnerable Machines

Discussed below are scanning methods used by an attacker to find vulnerable machines in a network:

- **Random Scanning**

  In this technique, the infected machine (an attacker's machine or a zombie) probes IP addresses randomly in the target network's IP range and checks their vulnerability. On finding a vulnerable machine, it hacks and attempts to infect the vulnerable machine by installing the same malicious code installed on it. This technique generates significant traffic because many compromised machines probe and check the same IP addresses. Malware propagates quickly in the initial stage, and the speed of propagation reduces as the number of new IP addresses available decreases with time.

- **Hit-list Scanning**

  Through scanning, an attacker first collects a list of potentially vulnerable machines and then creates a zombie army. Subsequently, the attacker scans the list to find a vulnerable machine. On finding one, the attacker installs malicious code on it and divides the list in half. The attacker continues to scan one half, whereas the other half is scanned by the newly compromised machine. This process keeps repeating, causing the number of compromised machines to increase exponentially. This technique ensures the installation of malicious code on all the potentially vulnerable machines in the hit list within a short time.

- **Topological Scanning**

  This technique uses the information obtained from an infected machine to find new vulnerable machines. An infected host checks for URLs in the hard drive of a machine that it wants to infect. Subsequently, it shortlists URLs and targets, and it checks their vulnerability. This technique yields accurate results, and its performance is similar to that of the hit-list scanning technique.

- **Local Subnet Scanning**

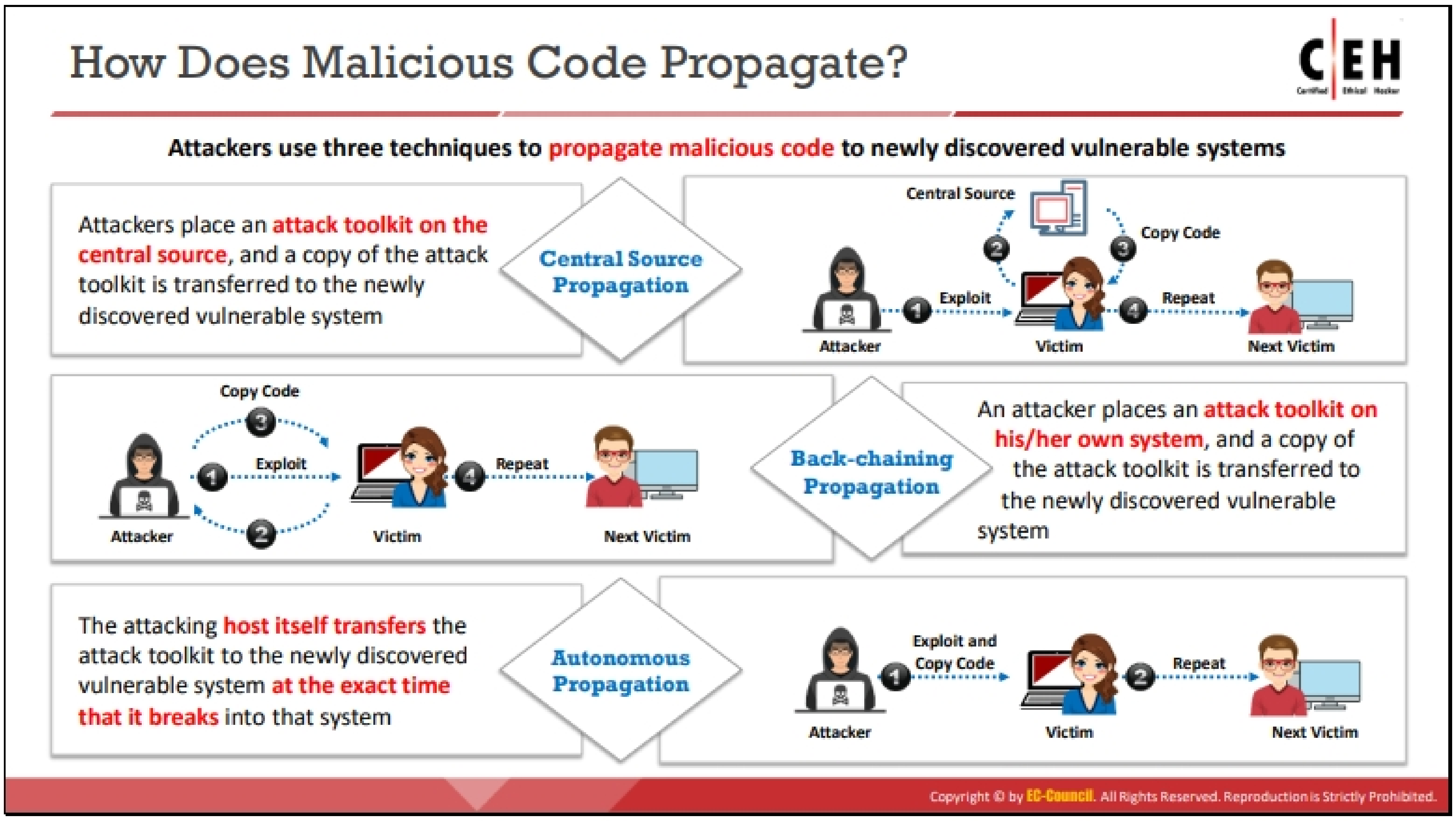
  In this technique, an infected machine searches for new vulnerable machines in its local network, behind a firewall, by using the information hidden in the local addresses. Attackers use this technique in combination with other scanning mechanisms.

- **Permutation Scanning**

  In this technique, attackers share a common pseudorandom permutation list of IP addresses of all machines. The list is created using a block cipher of 32 bits and a preselected key. If a compromised host is infected during either hit-list scanning or local subnet scanning, the list is scanned from immediately after the point of the compromised host to identify new targets. If a compromised host is infected during permutation scanning, scanning restarts from a random point. If an already infected machine is encountered, scanning restarts from a new random start point in the permutation list. The process of scanning stops when the compromised host consecutively encounters a predefined number of already infected machines and fails to find new targets. Thereafter, a new permutation key is generated to initiate a new scanning phase.

  Permutation scanning has the following advantages:

  - The reinfection of a target is avoided.
  - New targets are scanned at random, thereby ensuring a high scanning speed.



How Does Malicious Code Propagate?

Attackers use three techniques to propagate malicious code to newly discovered vulnerable systems

Attackers place an **attack toolkit on the central source**, and a copy of the attack toolkit is transferred to the newly discovered vulnerable system

Central Source Propagation

Central Source
Copy Code
Exploit
Repeat
Attacker
Victim
Next Victim

Copy Code
Exploit
Repeat
Attacker
Victim
Next Victim

Back-chaining Propagation

An attacker places an **attack toolkit on his/her own system**, and a copy of the attack toolkit is transferred to the newly discovered vulnerable system

The attacking **host itself transfers** the attack toolkit to the newly discovered vulnerable system **at the exact time that it breaks** into that system

Autonomous Propagation

Exploit and Copy Code
Repeat
Attacker
Victim
Next Victim

How Does Malicious Code Propagate?

Discussed below are three techniques used by an attacker to propagate malicious code and build attack networks:

- **Central Source Propagation**

  In this technique, the attacker places an attack toolkit on a central source and a copy of the attack toolkit is transferred to a newly discovered vulnerable system. Once the attacker finds a vulnerable machine, they instruct the central source to transfer a copy of the attack toolkit to the newly compromised machine, on which attack tools are automatically installed under management by a scripting mechanism. This initiates a new attack cycle, in which the newly infected machine searches for other vulnerable machines and repeats the process to install the attack toolkit. In general, this technique uses HTTP, FTP, and RPC protocols.

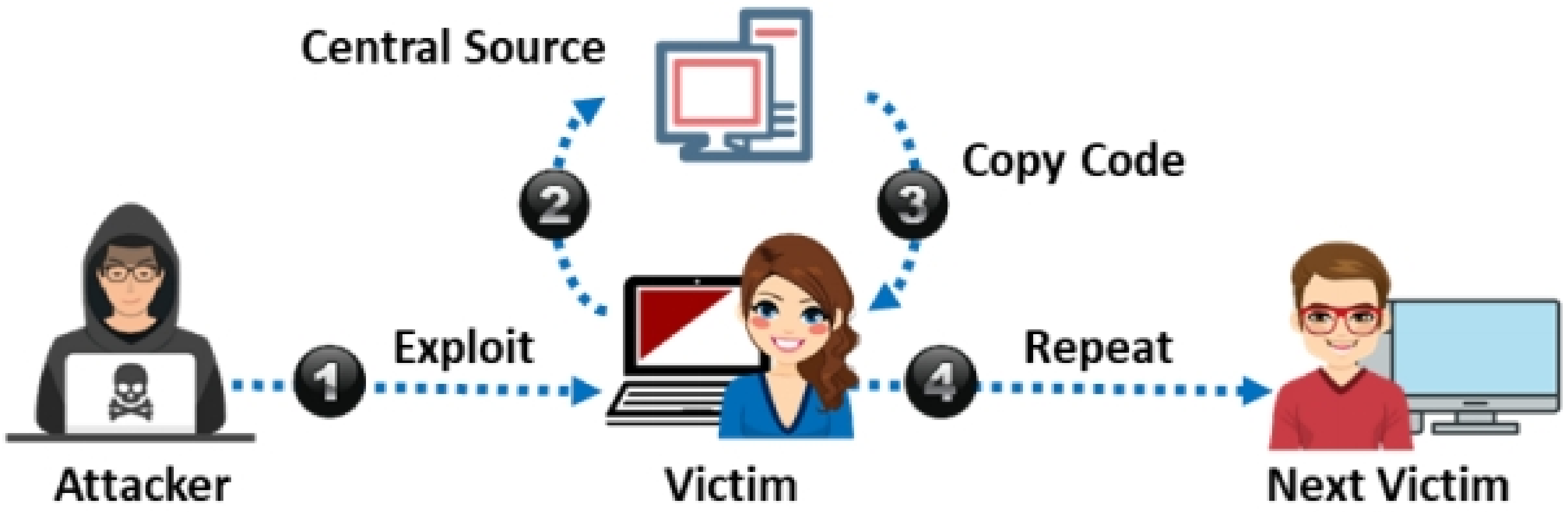

Central Source
Copy Code
Exploit
Repeat
Attacker
Victim
Next Victim

Figure 10.20: Central source propagation

- **Back-chaining Propagation**

  In this technique, the attacker places an attack toolkit on their own system, and a copy of the attack toolkit is transferred to a newly discovered vulnerable system. The attack tools installed on the attacking machine use some special methods to accept a connection from the compromised system and then transfer a file containing the attack tools to it. Simple port listeners containing a copy of this file or full intruder-installed web servers, both of which use the Trivial File Transfer Protocol (TFTP), support this back-channel file copy.

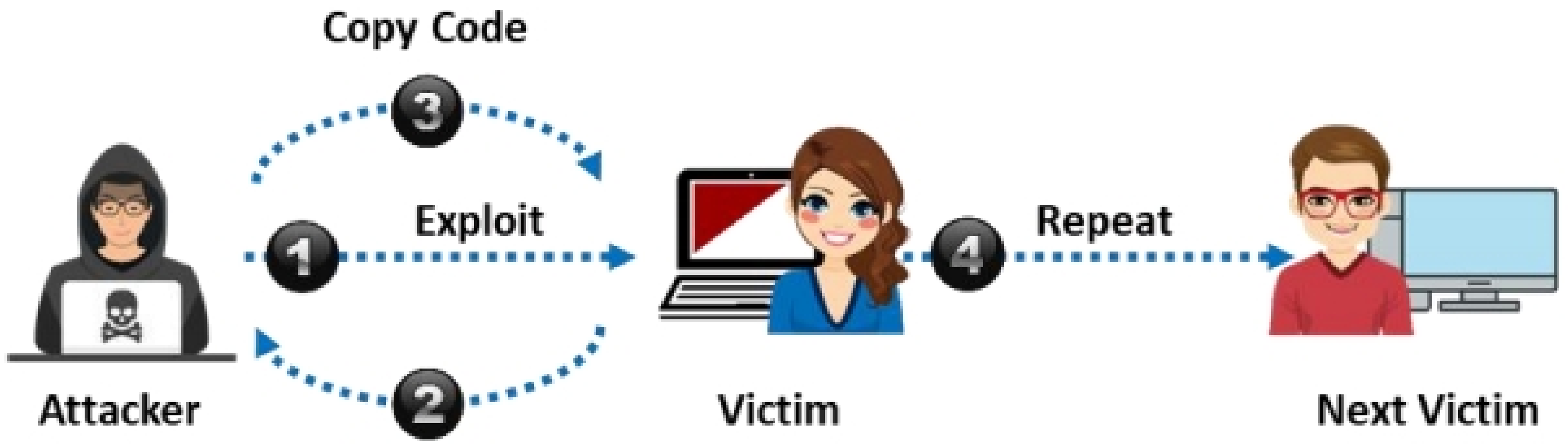


Figure 10.21: Back-chaining propagation

- **Autonomous Propagation**

  Unlike the previously discussed mechanisms, in which an external file source transfers the attack toolkit, in autonomous propagation, the attacking host itself transfers the attack toolkit to a newly discovered vulnerable system, exactly at the time it breaks into that system.

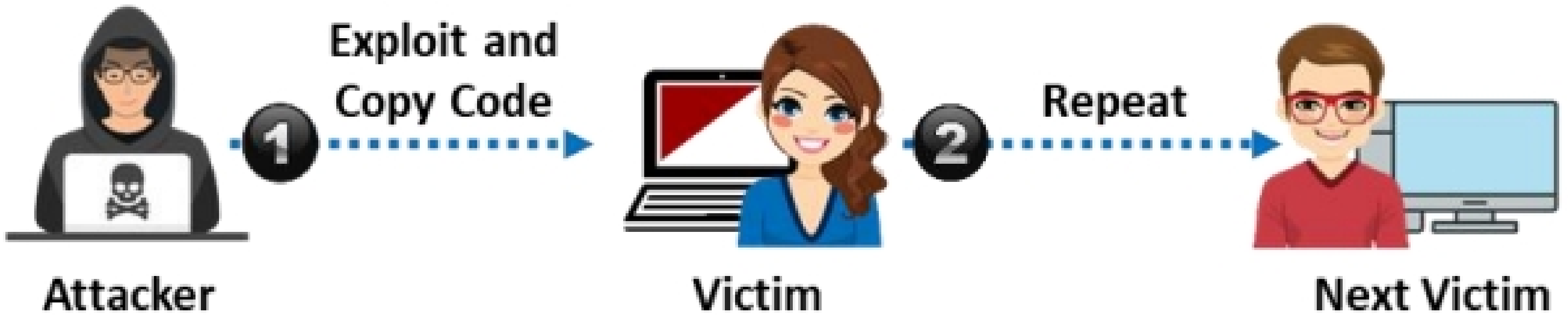
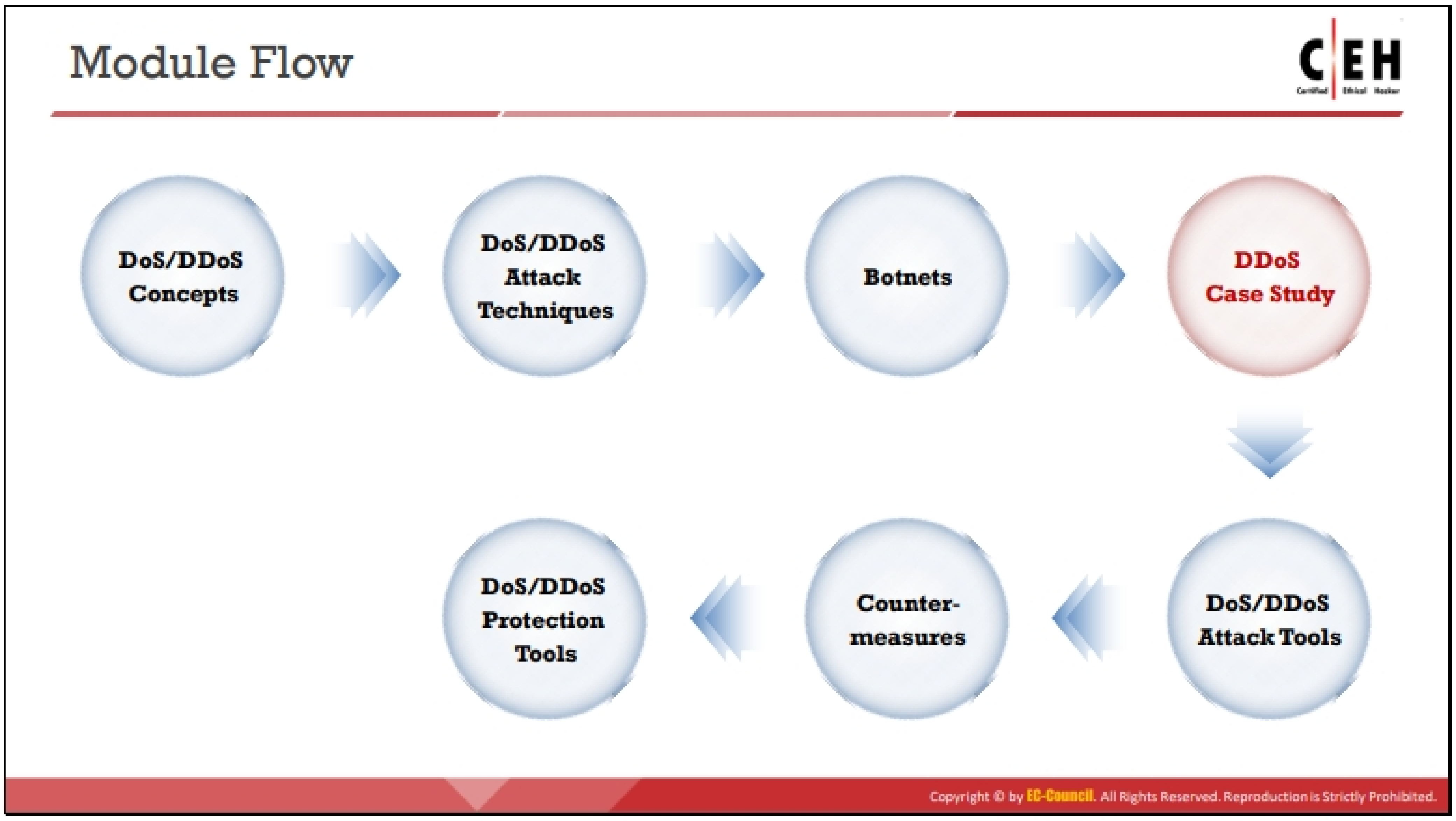


Figure 10.22: Autonomous Propagation

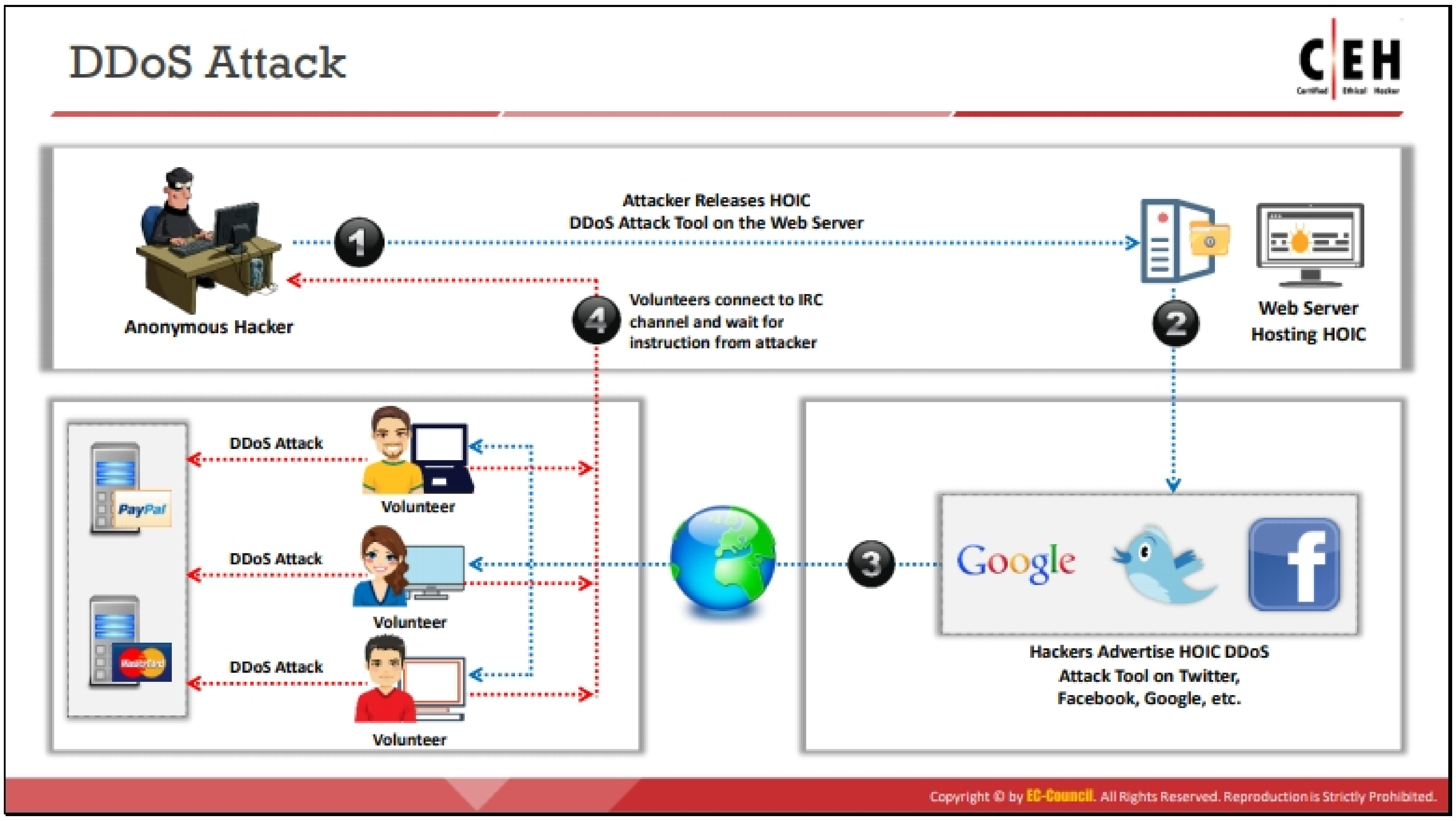


Module Flow

DoS/DDoS Concepts
DoS/DDoS Attack Techniques
Botnets
DDoS Case Study
DoS/DDoS Attack Tools
Counter-measures
DoS/DDoS Protection Tools
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

DDoS Case Study

DDoS attacks are sophisticated and complex attacks based on DoS and multiple distributed attack sources. In a DDoS attack, a large number of compromised computers (zombies) interrupt or suspend network services. This section presents a case study of a DDoS attack.

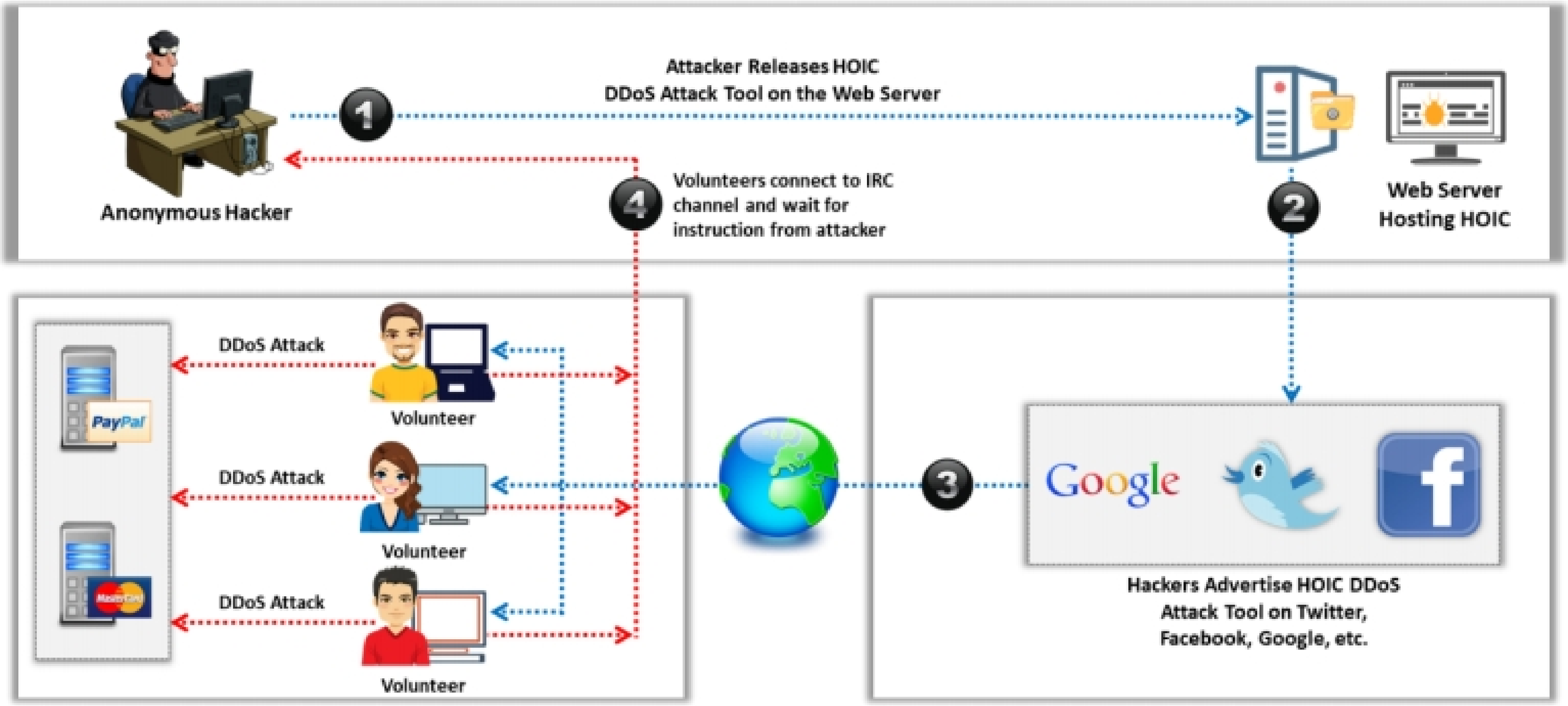


DDoS Attack

Attacker Releases HOIC
DDoS Attack Tool on the Web Server
1
Anonymous Hacker
4
Volunteers connect to IRC channel and wait for instruction from attacker
2
Web Server Hosting HOIC
DDoS Attack
Volunteer
DDoS Attack
Volunteer
DDoS Attack
Volunteer
PayPal
3
Google
Hackers Advertise HOIC DDoS Attack Tool on Twitter, Facebook, Google, etc.

DDoS Attack

In a DDoS attack, attackers use a group of compromised systems (bots or zombies) usually infected with Trojans to perform a DoS attack on a target system or network resource.

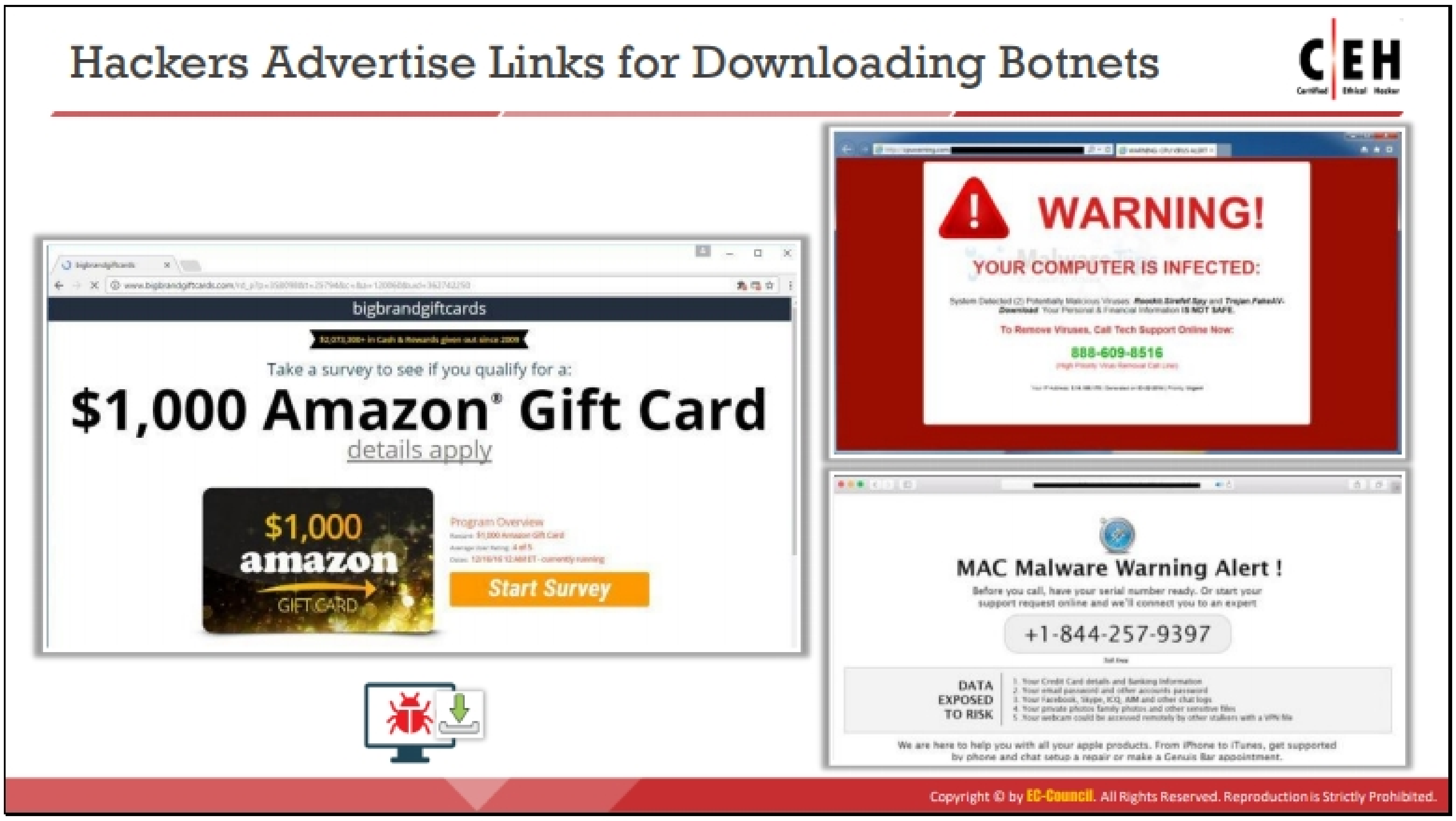


Attacker Releases HOIC
DDoS Attack Tool on the Web Server
1
Anonymous Hacker
4
Volunteers connect to IRC channel and wait for instruction from attacker
2
Web Server Hosting HOIC
DDoS Attack
Volunteer
DDoS Attack
Volunteer
DDoS Attack
Volunteer
PayPal
3
Google
Hackers Advertise HOIC DDoS Attack Tool on Twitter, Facebook, Google, etc.

Figure 10.23: DDoS attack scenario

As shown in the figure, an anonymous hacker hosts a High Orbit Ion Cannon (HOIC) DDoS attack tool on a web server they own or on a compromised web server. The hacker then advertises the HOIC DDoS attack tool on social networking sites or search engines such as Twitter, Facebook, and Google with a malicious download link.

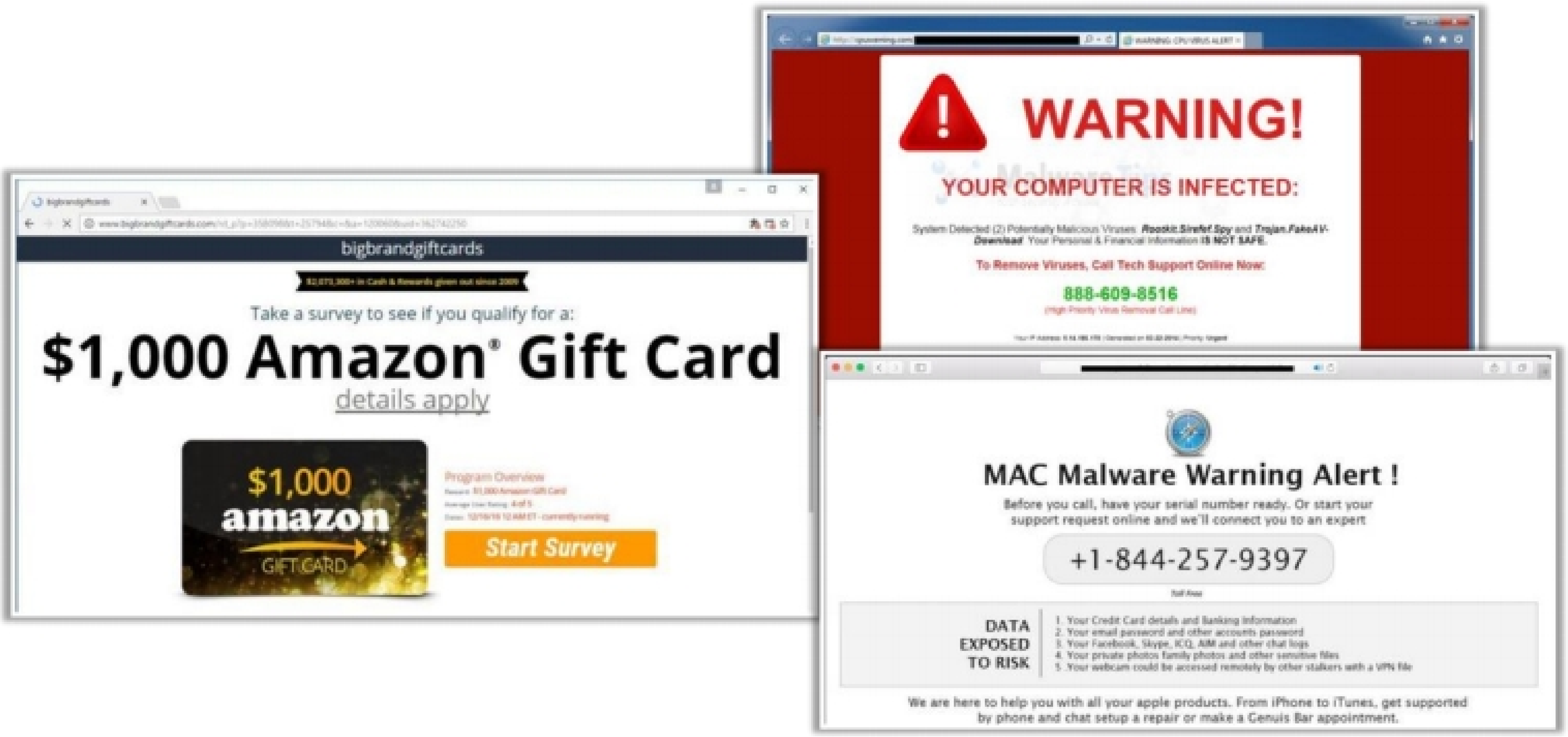
Users who desire to perform the DDoS attack may download the HOIC DDoS attack tool by clicking on the malicious download link provided by the hacker. These users are termed "volunteers." All the volunteers connect via an IRC channel to the anonymous hacker and await instructions to proceed further. The hacker instructs the volunteers to flood the target web server (e.g., PayPal, MasterCard, and PAYBACK) with multiple requests. On receiving instructions, the volunteers act accordingly. Consequently, the target server becomes overwhelmed and stops responding to requests from even legitimate users.



Hackers Advertise Links for Downloading Botnets

Hackers Advertise Links for Downloading Botnets

Hackers advertise botnets on various blogs, search engines, social networking sites, emails, and so on with download links. Hackers also use fake updates and security alerts to trick the victim into downloading the malware. The intention in doing so is to spread the botnet and increase the size of the attack network. This method of attack is very quick and effective. The below figure shows examples for ads hosted by hackers on the Internet to download botnets.



bigbrandgiftcards

Take a survey to see if you qualify for a:

$1,000 Amazon® Gift Card

details apply

Program Overview

WARNING!

YOUR COMPUTER IS INFECTED:

System Detected (2) Potentially Malicious Viruses: Rootkit.Sirefef.Spy and Trojan.FakeAV-Download. Your Personal & Financial Information IS NOT SAFE.

To Remove Viruses, Call Tech Support Online Now:

888-609-8516

(High Priority Virus Removal Call Line)

MAC Malware Warning Alert !

Before you call, have your serial number ready. Or start your support request online and we'll connect you to an expert

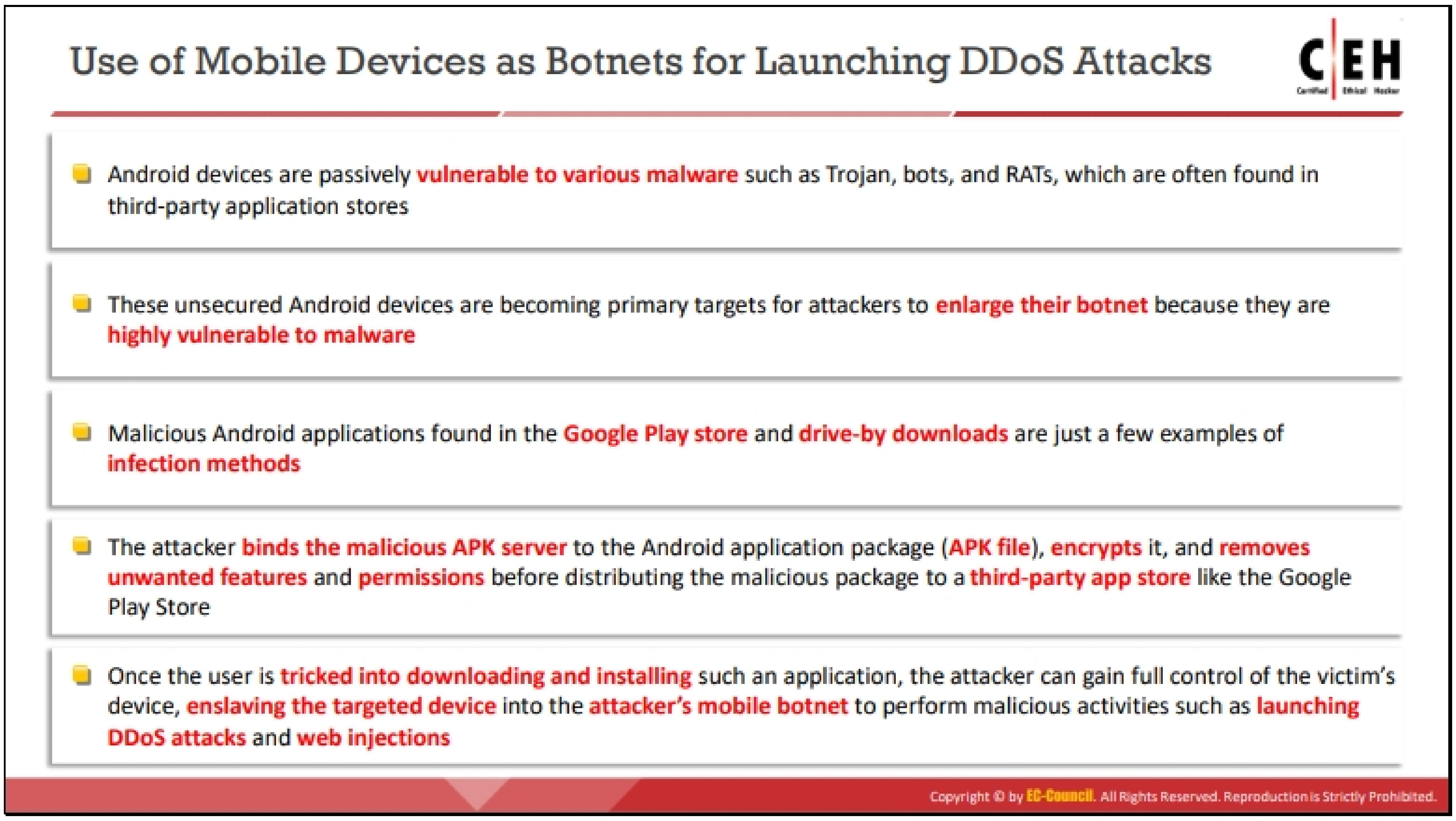
+1-844-257-9397

DATA EXPOSED TO RISK

1. Your Credit Card details and Banking Information
2. Your email password and other accounts password
3. Your Facebook, Skype, ICQ, AIM and other chat logs
4. Your private photos family photos and other sensitive files
5. Your webcam could be accessed remotely by other stalkers with a VPN file

We are here to help you with all your apple products. From iPhone to iTunes, get supported by phone and chat setup a repair or make a Genuis Bar appointment.

Figure 10.24: Advertisements with links to download botnets

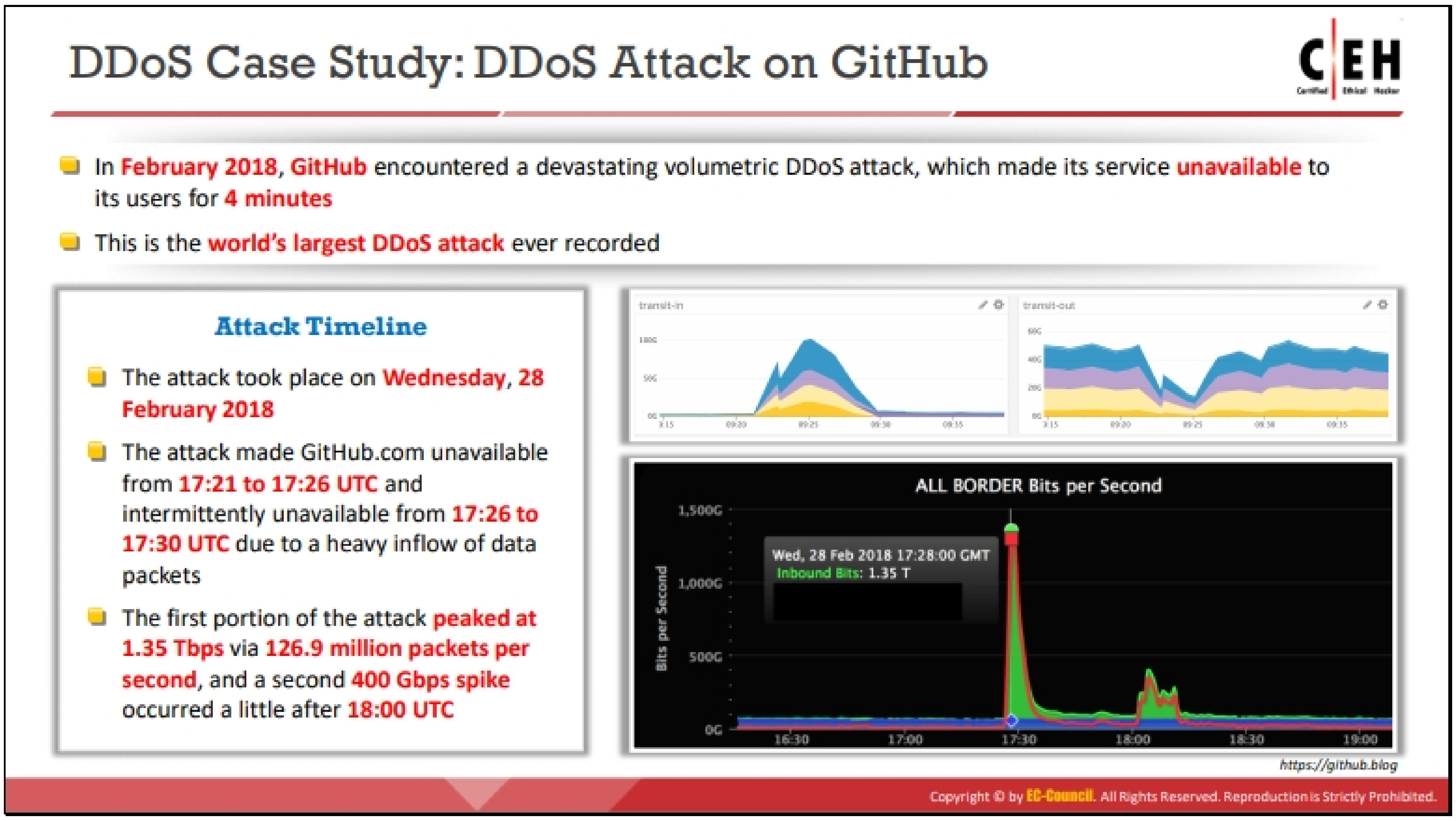


Use of Mobile Devices as Botnets for Launching DDoS Attacks

- Android devices are passively **vulnerable to various malware** such as Trojan, bots, and RATs, which are often found in third-party application stores
- These unsecured Android devices are becoming primary targets for attackers to **enlarge their botnet** because they are **highly vulnerable to malware**
- Malicious Android applications found in the **Google Play store** and **drive-by downloads** are just a few examples of **infection methods**
- The attacker **binds the malicious APK server** to the Android application package (**APK file**), **encrypts** it, and **removes unwanted features** and **permissions** before distributing the malicious package to a **third-party app store** like the Google Play Store
- Once the user is **tricked into downloading and installing** such an application, the attacker can gain full control of the victim's device, **enslaving the targeted device** into the **attacker's mobile botnet** to perform malicious activities such as **launching DDoS attacks** and **web injections**

Use of Mobile Devices as Botnets for Launching DDoS Attacks

Android devices are passively vulnerable to various malware such as Trojans, bots, Remote Access Trojans (RATs), and so on, which are often found in third-party application stores. These unsecured android devices are becoming the primary targets for attackers to enlarge their botnet network because they are highly vulnerable to malware. Malicious android applications found in the Google Play Store and as drive-by downloads are examples for infection methods. The attacker binds a malicious server to the android application package (APK) file, encrypts it, and removes unwanted features and permissions before distributing the malicious package to a third-party app store such as Google Play Store. Once the victims are tricked into downloading and installing such applications, the victim's device is taken over by the attacker and integrated into the attacker's mobile botnet to perform malicious activities such as DDoS attacks and web injections.



DDoS Case Study: DDoS Attack on GitHub

- In **February 2018**, **GitHub** encountered a devastating volumetric DDoS attack, which made its service **unavailable** to its users for **4 minutes**
- This is the **world's largest DDoS attack** ever recorded

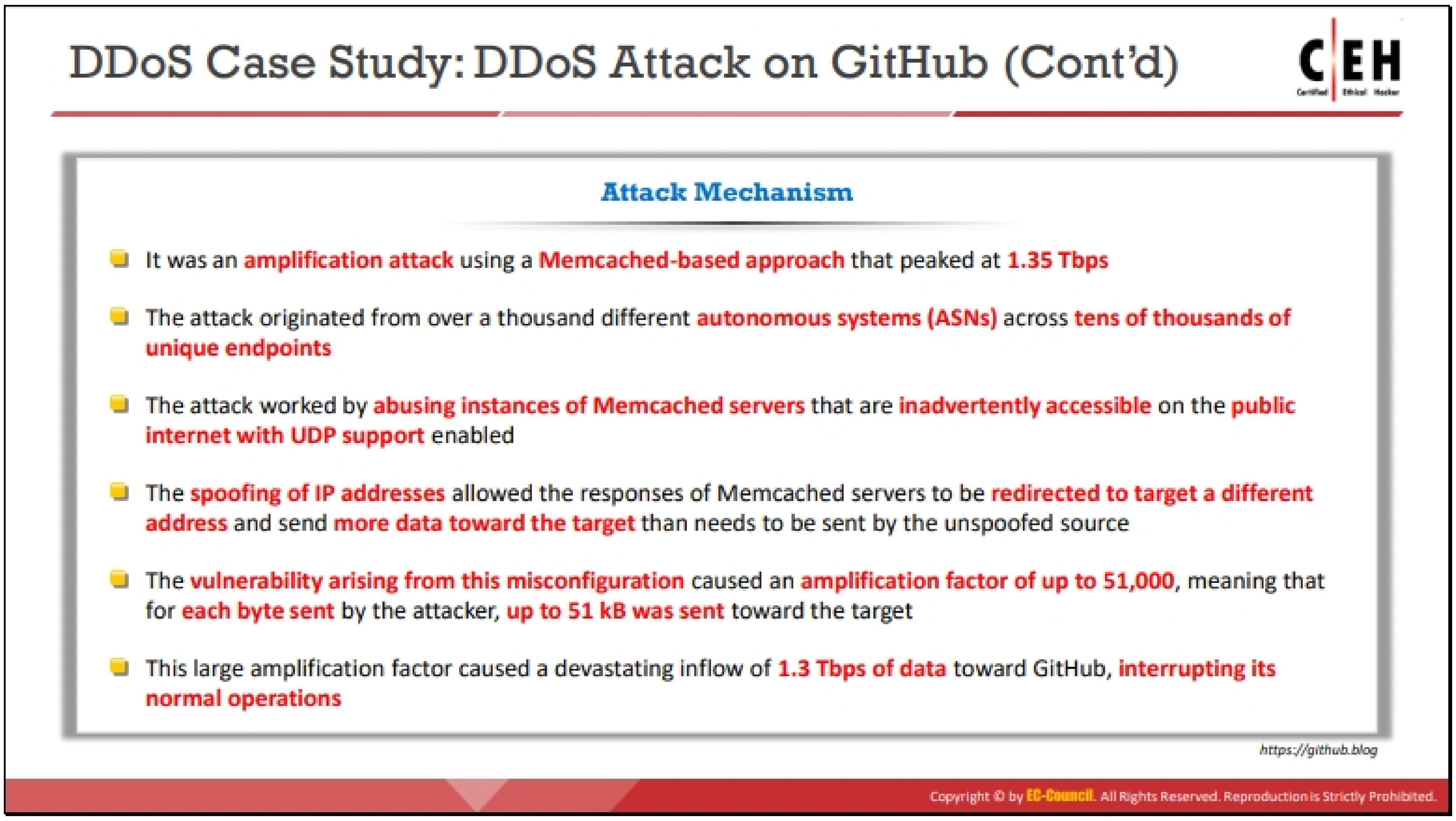
Attack Timeline

- The attack took place on **Wednesday, 28 February 2018**
- The attack made GitHub.com unavailable from **17:21 to 17:26 UTC** and intermittently unavailable from **17:26 to 17:30 UTC** due to a heavy inflow of data packets
- The first portion of the attack **peaked at 1.35 Tbps** via **126.9 million packets per second**, and a second **400 Gbps spike** occurred a little after **18:00 UTC**

transit-in
100G
50G
0G
09:15
09:20
09:25
09:30
09:35
transit-out
60G
40G
20G
0G
09:15
09:20
09:25
09:30
09:35
ALL BORDER Bits per Second
Bits per Second
1,500G
1,000G
500G
0G
16:30
17:00
17:30
18:00
18:30
19:00
Wed, 28 Feb 2018 17:28:00 GMT
Inbound Bits: 1.35 T

[redacted]

https://github.blog

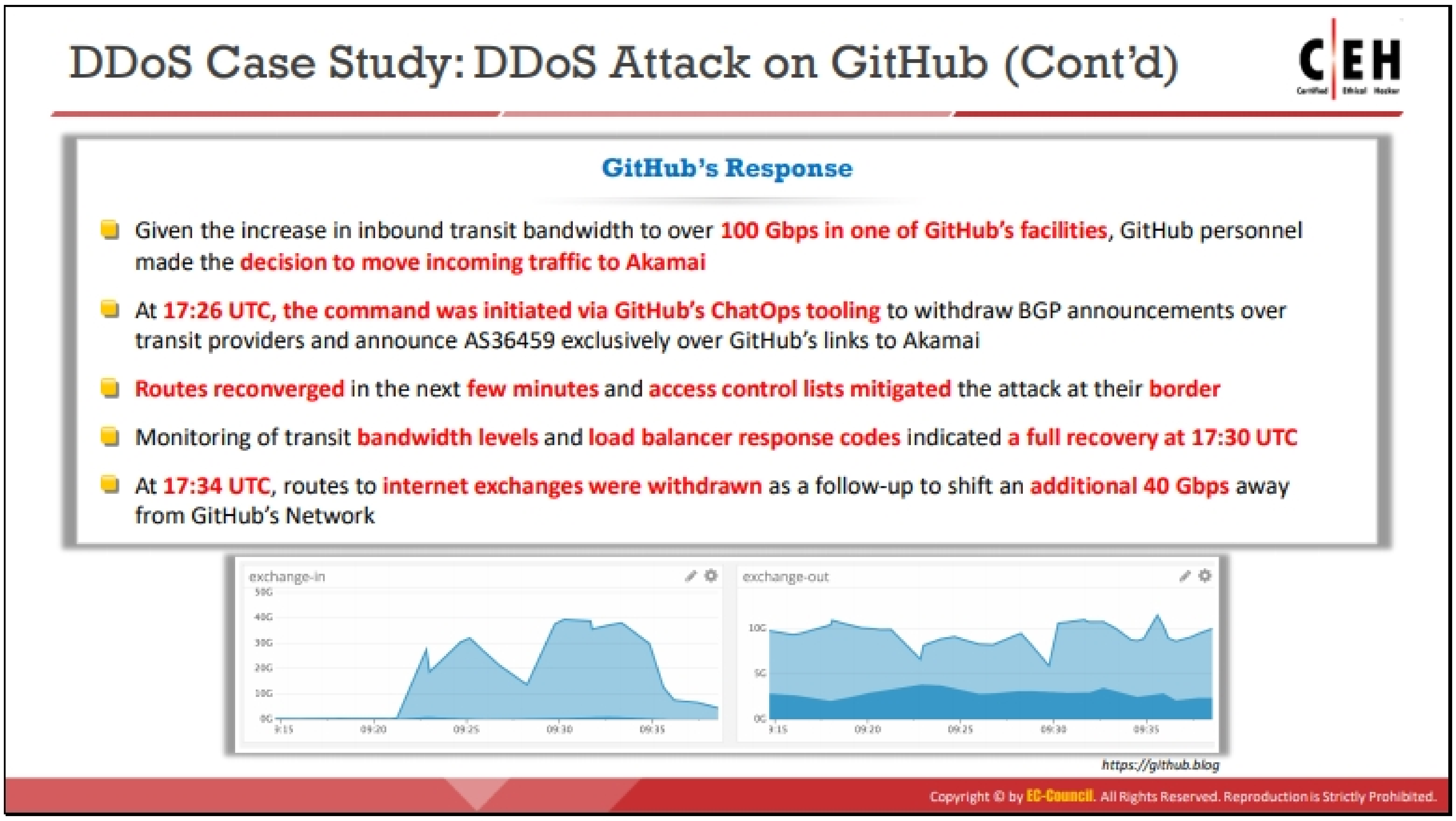


DDoS Case Study: DDoS Attack on GitHub (Cont'd)

Attack Mechanism

- It was an **amplification attack** using a **Memcached-based approach** that peaked at **1.35 Tbps**
- The attack originated from over a thousand different **autonomous systems (ASNs)** across **tens of thousands of unique endpoints**
- The attack worked by **abusing instances of Memcached servers** that are **inadvertently accessible** on the **public internet with UDP support** enabled
- The **spoofing of IP addresses** allowed the responses of Memcached servers to be **redirected to target a different address** and send **more data toward the target** than needs to be sent by the unspoofed source
- The **vulnerability arising from this misconfiguration** caused an **amplification factor of up to 51,000**, meaning that for **each byte sent** by the attacker, **up to 51 kB was sent** toward the target
- This large amplification factor caused a devastating inflow of **1.3 Tbps of data** toward GitHub, **interrupting its normal operations**

https://github.blog



DDoS Case Study: DDoS Attack on GitHub (Cont'd)

GitHub's Response

- Given the increase in inbound transit bandwidth to over **100 Gbps in one of GitHub's facilities**, GitHub personnel made the **decision to move incoming traffic to Akamai**
- At **17:26 UTC, the command was initiated via GitHub's ChatOps tooling** to withdraw BGP announcements over transit providers and announce AS36459 exclusively over GitHub's links to Akamai
- **Routes reconverged** in the next **few minutes** and **access control lists mitigated** the attack at their **border**
- Monitoring of transit **bandwidth levels** and **load balancer response codes** indicated **a full recovery at 17:30 UTC**
- At **17:34 UTC**, routes to **internet exchanges were withdrawn** as a follow-up to shift an **additional 40 Gbps** away from GitHub's Network

https://github.blog

DDoS Case Study: DDoS Attack on GitHub

Source: *https://github.blog*

GitHub is a renowned open-source cloud platform used as a repository by many companies, businesses, and researchers with wide areas of interest. In February 2018, GitHub encountered a devastating DDoS attack, which made its service unavailable to users for 4 min. This was the world's largest DDoS attack ever recorded.

- **Attack Timeline**

  The DDoS attack occurred on Wednesday, February 28, 2018. This volumetric DDoS attack made GitHub.com unavailable from 17:21 to 17:26 UTC and intermittently unavailable from 17:26 to 17:30 UTC owing to a heavy inflow of data packets.

  At 17:21 UTC, GitHub's network monitoring system detected an anomaly in the ratio of ingress to egress traffic, and they notified the on-call engineer and others. The below figure shows inbound versus outbound throughput over transit links.

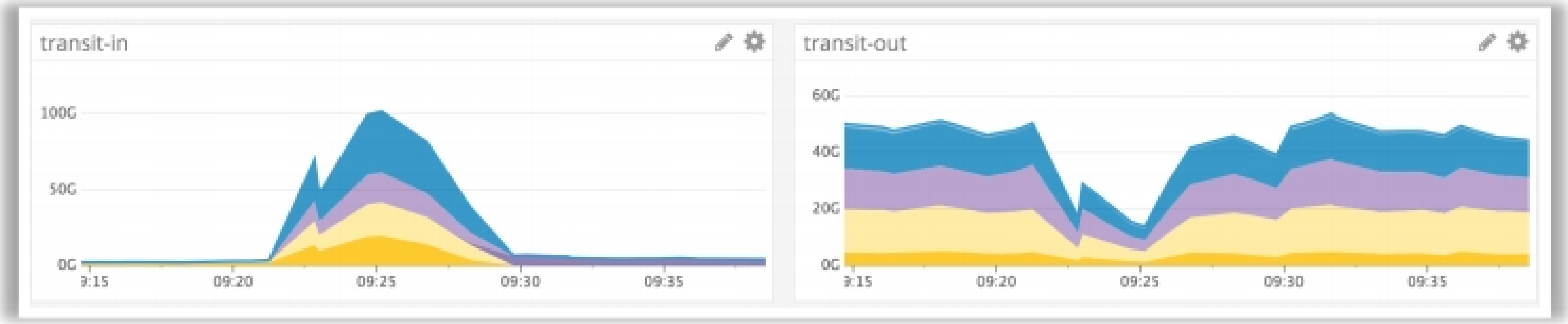


Figure 10.25: Inbound versus outbound throughput over transit links

The first portion of the attack peaked at 1.35 Tbps via 126.9 million packets per second (pps), and a second 400 Gbps spike occurred a little after 18:00 UTC. The below figure, provided by Akamai, shows inbound traffic in bits per second (bps).

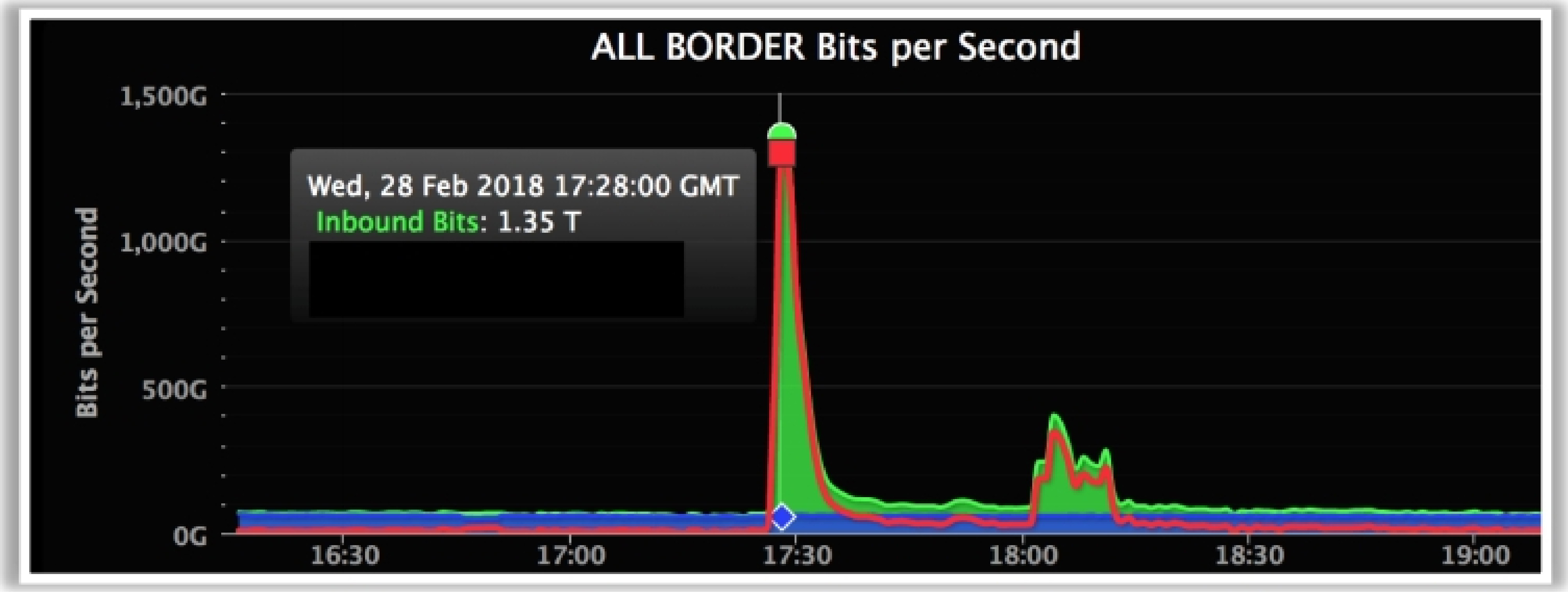


Figure 10.26: Inbound traffic in bits per second

- **Attack Mechanism**

  This attack was an amplification attack using a Memcached-based approach that peaked at 1.35 Tbps. The attack originated from over a thousand different autonomous system numbers (ASNs) across tens of thousands of unique endpoints. The attack worked by abusing Memcached instances that were inadvertently accessible on the public Internet with UDP support enabled. The spoofing of IP addresses allowed Memcached's responses to be targeted against another address and more data than necessary to be sent toward the target by the unspoofed source. The vulnerability due to this misconfiguration caused an amplification factor of up to 51,000, implying that up to 51 KB was sent toward the target for each byte sent by the attacker. This large amplification factor caused the devastating inflow of 1.3 Tbps data towards GitHub, interrupting its normal operations.

- **GitHub's Response**

  Given the increase in inbound transit bandwidth to over 100 Gbps in one of GitHub's facilities, GitHub personnel decided to move the incoming traffic to Akamai. At 17:26 UTC, a command was initiated via GitHub's ChatOps tooling to withdraw Border Gateway Protocol (BGP) announcements over transit providers and announce AS36459, the autonomous system belonging to GitHub, exclusively over GitHub's links to Akamai.

  Routes reconverged in the next few minutes, and access control lists mitigated the attack at their border. The monitoring of transit bandwidth levels and load balancer response codes indicated a full recovery at 17:30 UTC. At 17:34 UTC, routes to Internet exchanges were withdrawn as a follow-up to shift an additional 40 Gbps away from GitHub's network.

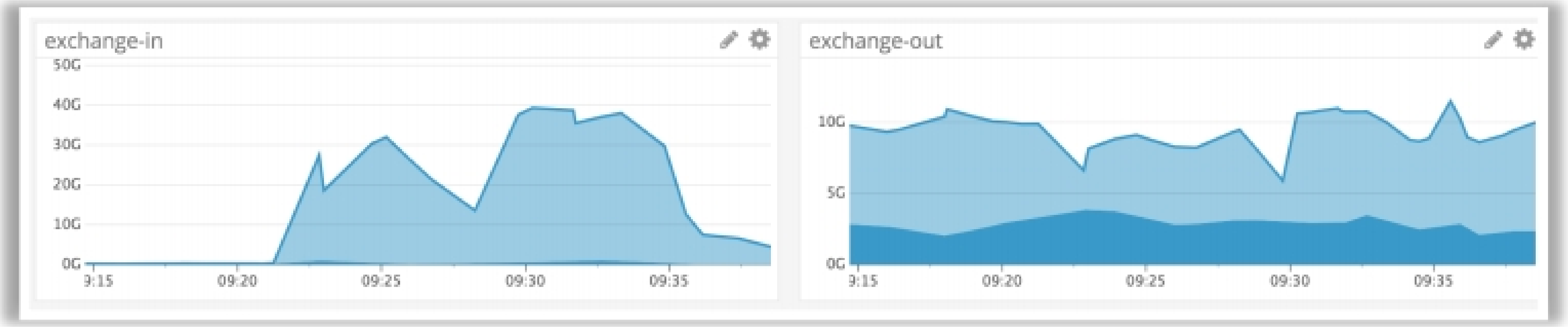
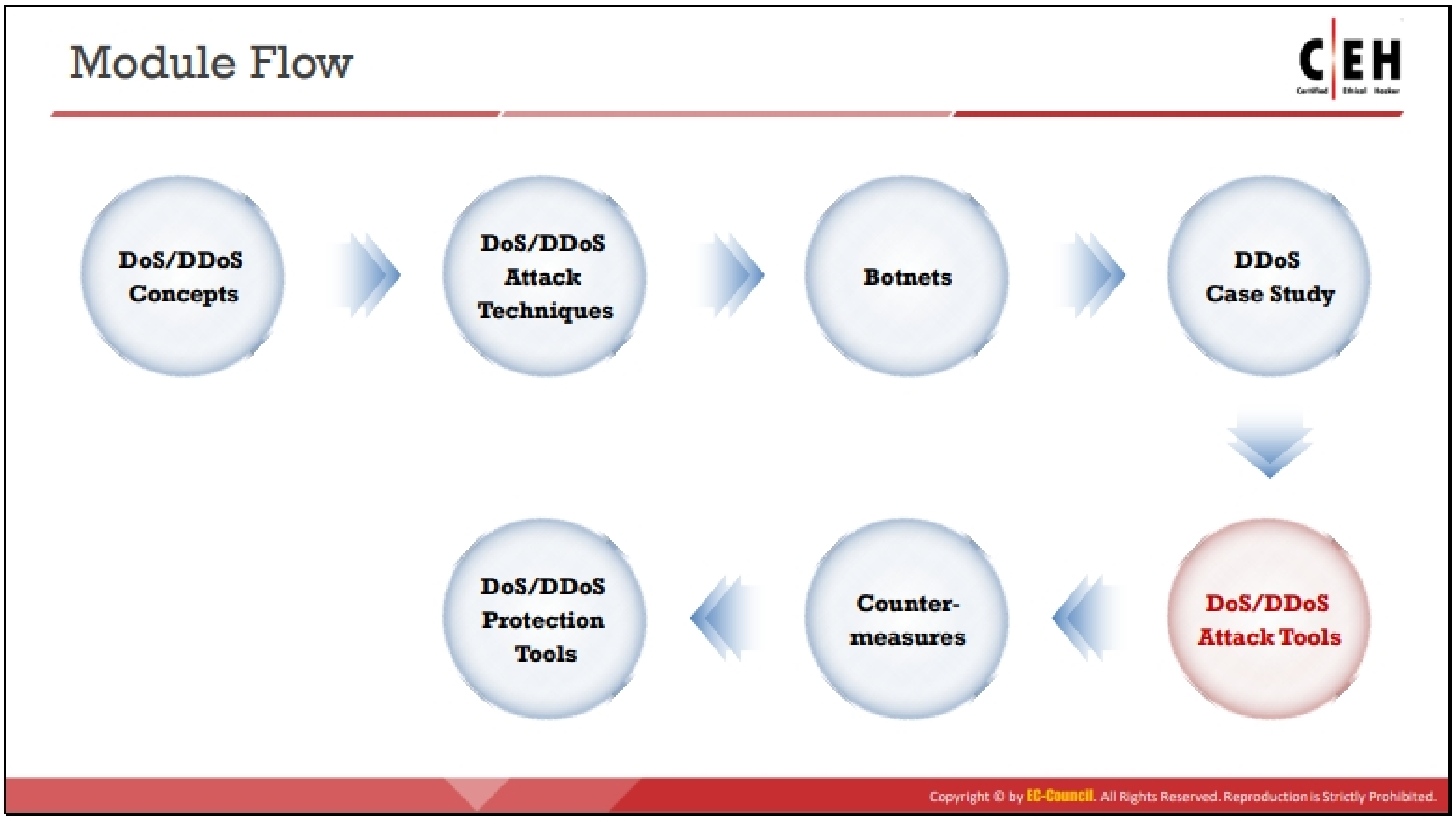


Figure 10.27: Inbound versus outbound throughput over transit links

After this incident, GitHub revealed that they were investigating the use of their monitoring infrastructure to automate DDoS mitigation providers and would continue to measure response times to similar incidents with a goal of reducing the mean time to recovery (MTTR).



Module Flow

DoS/DDoS Concepts

DoS/DDoS Attack Techniques

Botnets

DDoS Case Study

DoS/DDoS Attack Tools

Counter-measures

DoS/DDoS Protection Tools

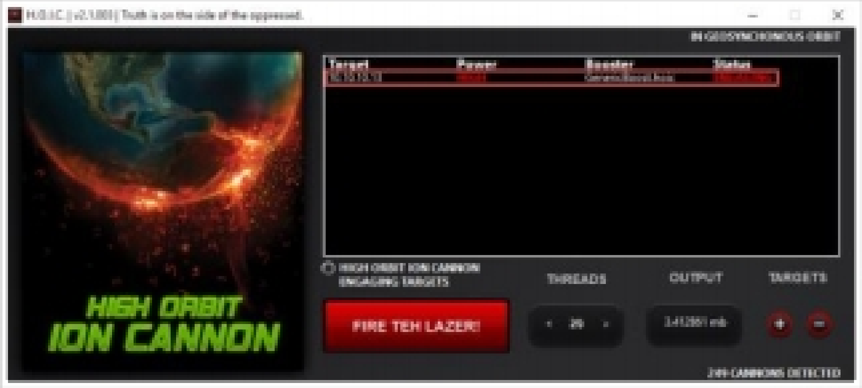
DoS/DDoS Attack Tools

This section deals with various DoS/DDoS attack tools used to take over a single or multiple network system to exhaust their computing resources or render them unavailable to their intended users.

DoS/DDoS Attack Tools

High Orbit Ion Cannon (HOIC)

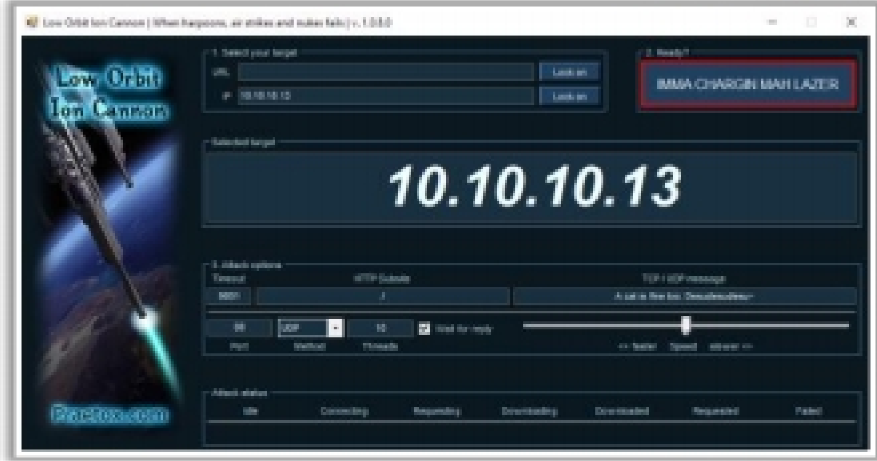
HOIC carries out a DDoS to attack **any IP address** with a user selected port and a user selected protocol



https://sourceforge.net

Low Orbit Ion Cannon (LOIC)

LOIC can be used on a **target site** to flood the server with TCP packets, UDP packets, or HTTP requests with the intention of **disrupting the service** of a particular host



https://sourceforge.net

DoS/DDoS Attack Tools

- XOIC (*http://anonhacktivism.blogspot.com*)
- HULK (*https://siberianlaika.ru*)
- Tor's Hammer (*https://sourceforge.net*)
- Slowloris (*https://github.com*)
- PyLoris (*https://sourceforge.net*)
- R-U-Dead-Yet (*https://sourceforge.net*)

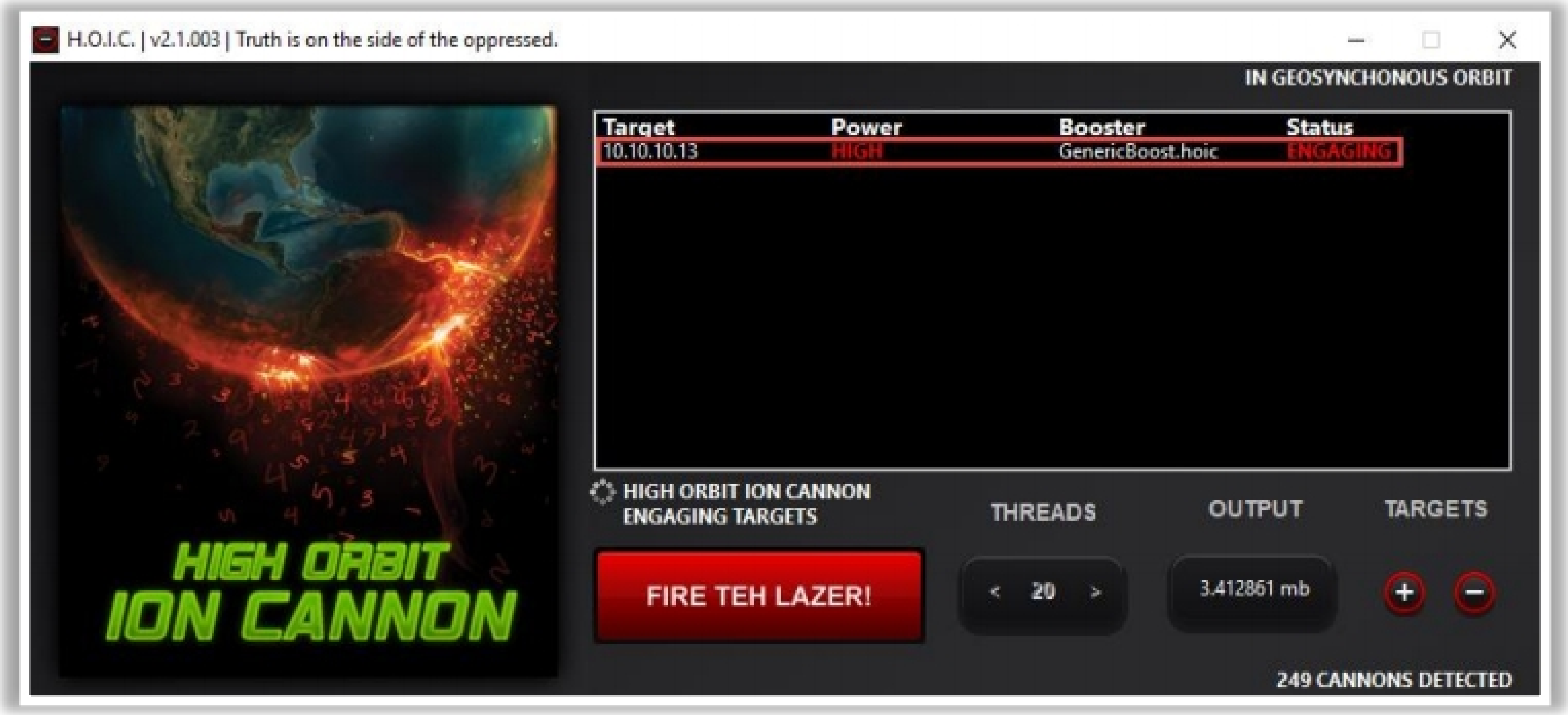
DoS/DDoS Attack Tools

- **High Orbit Ion Cannon (HOIC)**

  Source: *https://sourceforge.net*

  HOIC is a network stress and DoS/DDoS attack application written in BASIC language. It is designed to attack up to 256 target URLs simultaneously. It sends HTTP POST and GET requests to a computer that uses lulz-inspired GUIs. Its features are summarized as follows:

  - High-speed multi-threaded HTTP flooding
  - Simultaneous flooding of up to 256 websites
  - Built-in scripting system to allow the deployment of "boosters," which are scripts designed to thwart DDoS countermeasures and increase DoS output
  - Portability to Linux/Mac with a few bug fixes
  - Ability to select the number of threads in an ongoing attack
  - Ability to throttle attacks individually with three settings: LOW, MEDIUM, and HIGH



H.O.I.C. | v2.1.003 | Truth is on the side of the oppressed.

Target	Power	Booster	Status
10.10.10.13	HIGH	GenericBoost.hoic	ENGAGING

THREADS: 20

OUTPUT: 3.412861 mb

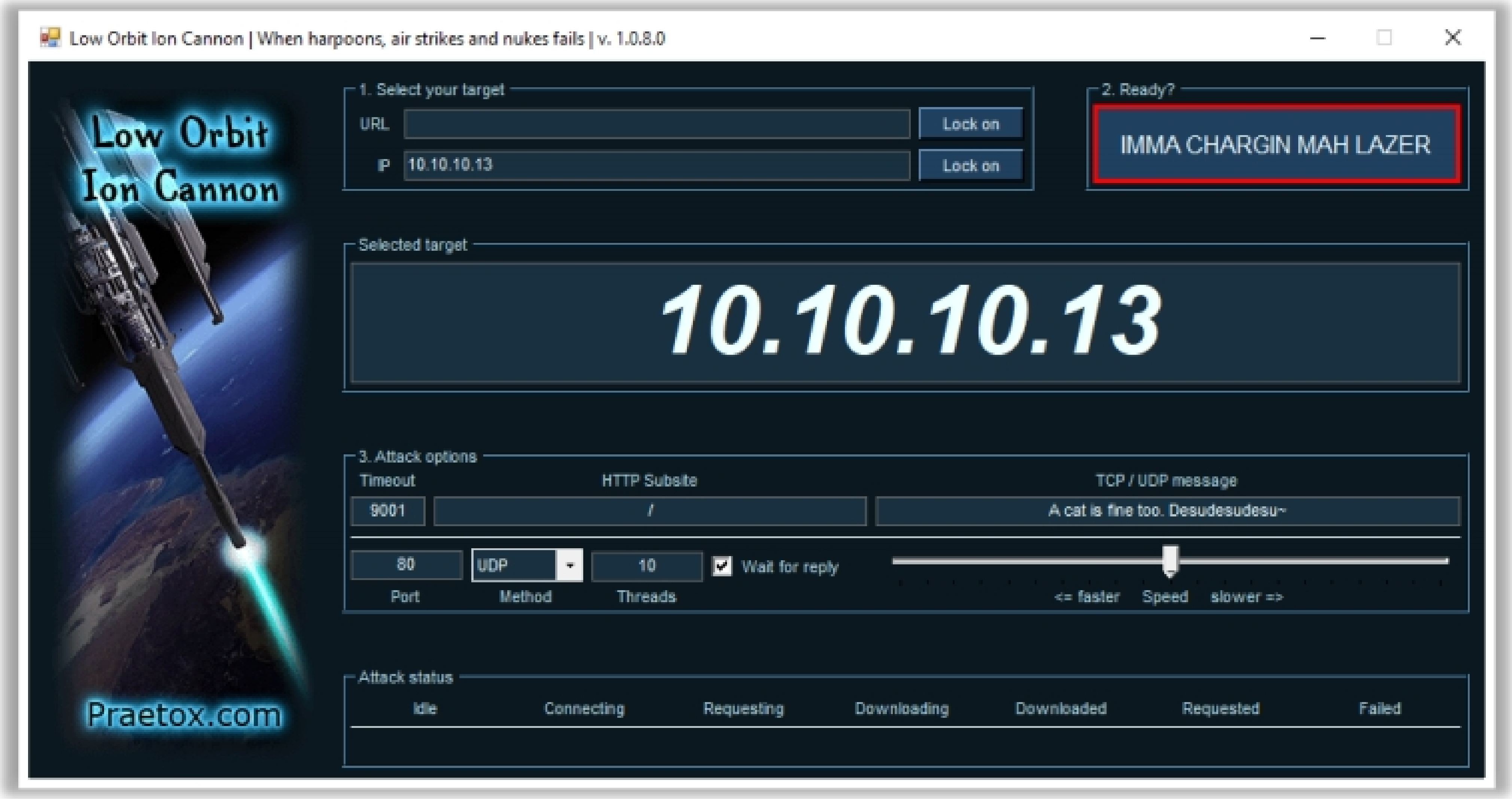
IN GEOSYNCHONOUS ORBIT
HIGH ORBIT ION CANNON
HIGH ORBIT ION CANNON ENGAGING TARGETS
FIRE TEH LAZER!
<
>
TARGETS
+
−
249 CANNONS DETECTED

Figure 10.28: Screenshot of HOIC DoS attack tool

- **Low Orbit Ion Cannon (LOIC)**

  Source: *https://sourceforge.net*

  LOIC is a network stress testing and DoS attack application. LOIC attacks can be called application-based DOS attacks because they primarily target web applications. LOIC can be used on a target site to flood the server with TCP packets, UDP packets, or HTTP requests with the intention of disrupting the service.



Low Orbit Ion Cannon | When harpoons, air strikes and nukes fails | v. 1.0.8.0

1. Select your target
URL:
IP: 10.10.10.13

Selected target: 10.10.10.13

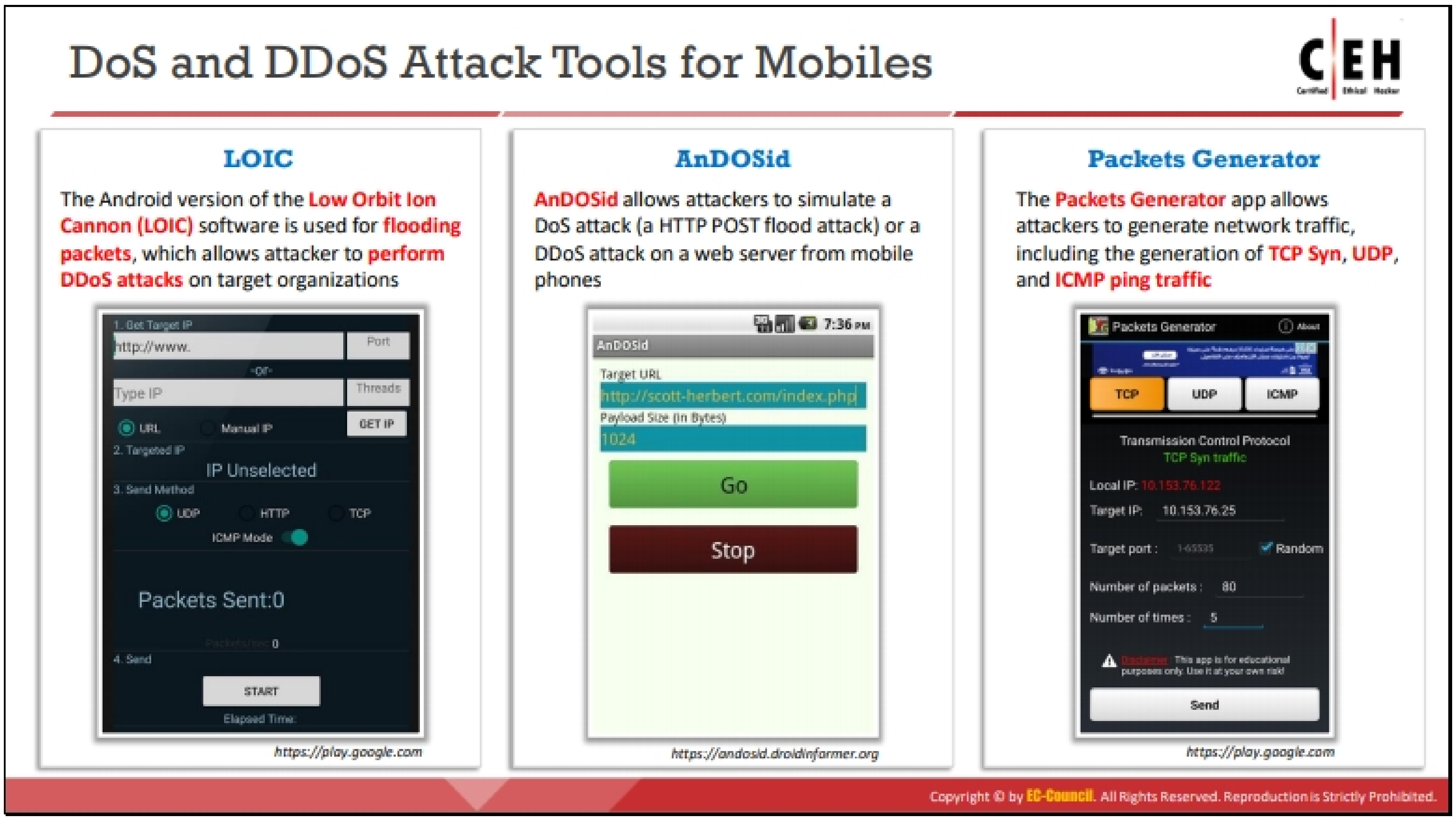
3. Attack options
Timeout: 9001
HTTP Subsite: /
TCP / UDP message: A cat is fine too. Desudesudesu~
Port: 80
Method: UDP
Threads: 10
☑ Wait for reply

Low Orbit Ion Cannon
Praetox.com
Lock on
Lock on
2. Ready?
IMMA CHARGIN MAH LAZER
<= faster Speed slower =>
Attack status
Idle
Connecting
Requesting
Downloading
Downloaded
Requested
Failed

Figure 10.29: Screenshot of LOIC DoS attack tool

The following are some of the additional DoS/DDoS attack tools:

- XOIC (*http://anonhacktivism.blogspot.com*)
- HULK (*https://siberianlaika.ru*)
- Tor's Hammer (*https://sourceforge.net*)
- Slowloris (*https://github.com*)
- PyLoris (*https://sourceforge.net*)
- R-U-Dead-Yet (*https://sourceforge.net*)



DoS and DDoS Attack Tools for Mobiles

LOIC

The Android version of the **Low Orbit Ion Cannon (LOIC)** software is used for **flooding packets**, which allows attacker to **perform DDoS attacks** on target organizations

https://play.google.com

AnDOSid

AnDOSid allows attackers to simulate a DoS attack (a HTTP POST flood attack) or a DDoS attack on a web server from mobile phones

https://andosid.droidinformer.org

Packets Generator

The **Packets Generator** app allows attackers to generate network traffic, including the generation of **TCP Syn**, **UDP**, and **ICMP ping traffic**

https://play.google.com

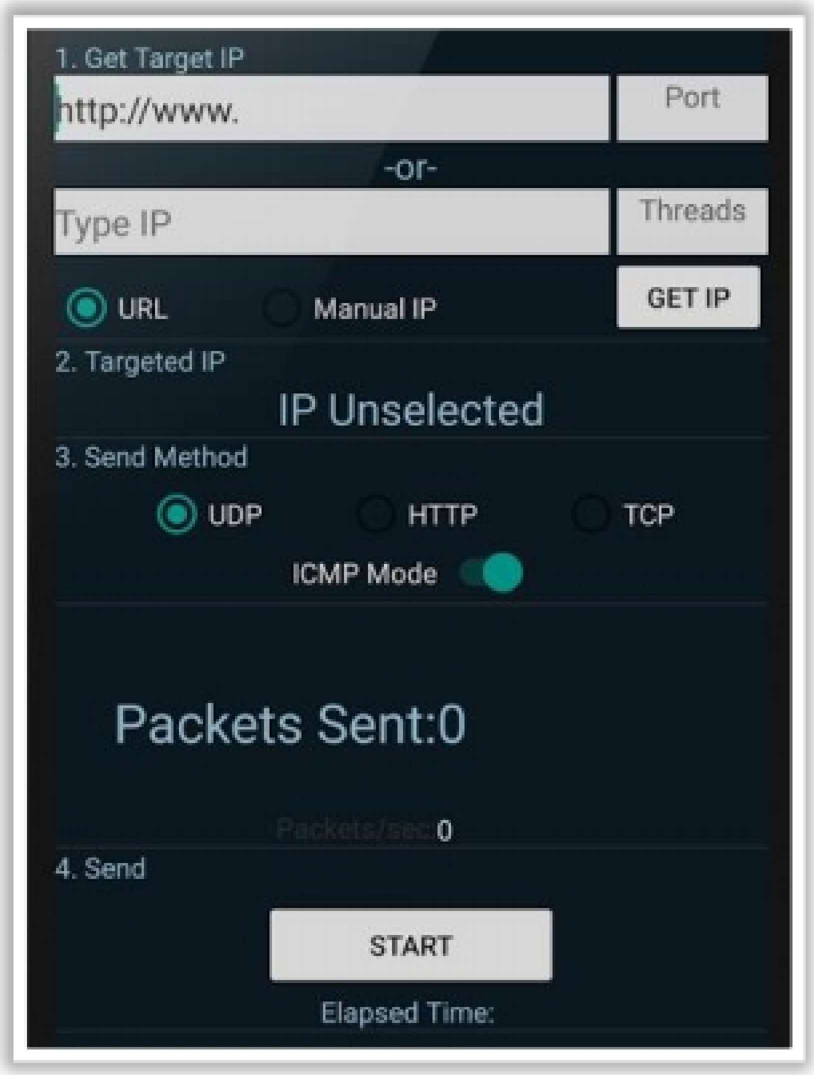
AnDOSid
Target URL
http://scott-herbert.com/index.php
Payload Size (in Bytes)
1024
Go
Stop
Packets Generator
TCP
UDP
ICMP
Transmission Control Protocol
TCP Syn traffic
Local IP: 10.153.76.122
Target IP: 10.153.76.25
Target port :
Random
Number of packets : 80
Number of times : 5
Send

DoS/DDoS Attack Tools for Mobiles

- **LOIC**

  Source: *https://play.google.com*

  The Android version of LOIC software is used for flooding packets, which allows the attacker to perform a DDoS attack on the target organization. This application can perform UPD, HTTP, or TCP flood attacks.



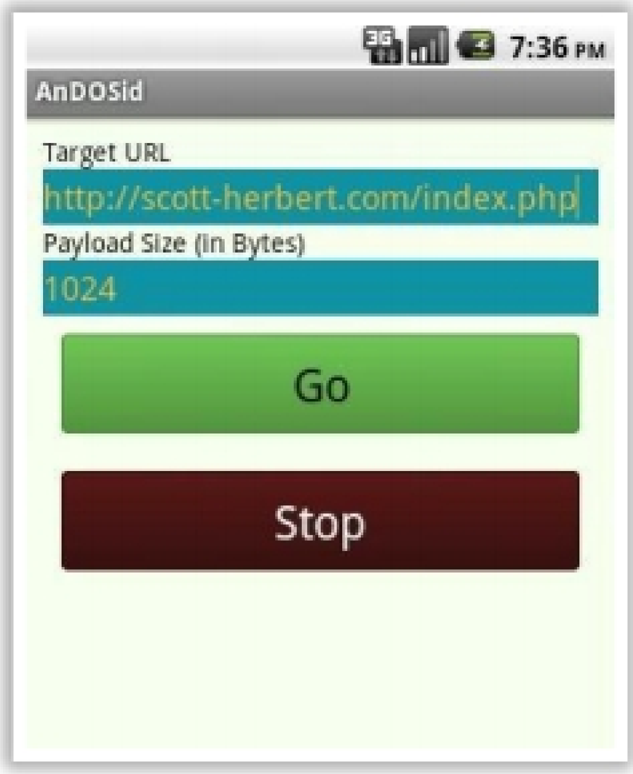
1. Get Target IP
http://www.
Port
-or-
Type IP
Threads
● URL
○ Manual IP
GET IP
2. Targeted IP
IP Unselected
3. Send Method
● UDP
○ HTTP
○ TCP
ICMP Mode
Packets Sent:0
Packets/sec: 0
4. Send
START
Elapsed Time:

Figure 10.30: Screenshot of LOIC DoS attack tool for mobile

- **AnDOSid**

  Source: *https://andosid.droidinformer.org*

  AnDOSid allows the attacker to simulate a DoS attack (an HTTP POST flood attack to be precise) and DDoS attack on a web server from mobile phones.



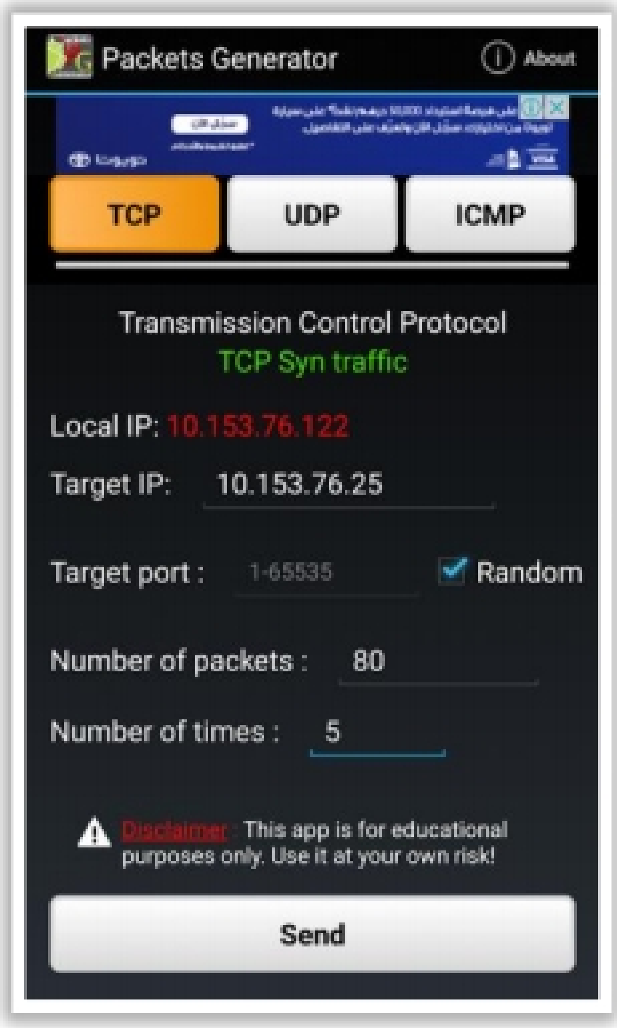
Target URL
http://scott-herbert.com/index.php
Payload Size (in Bytes)
1024

Figure 10.31: Screenshot of AnDOSid DoS attack tool for mobile

- **Packets Generator**

  Source: https://play.google.com

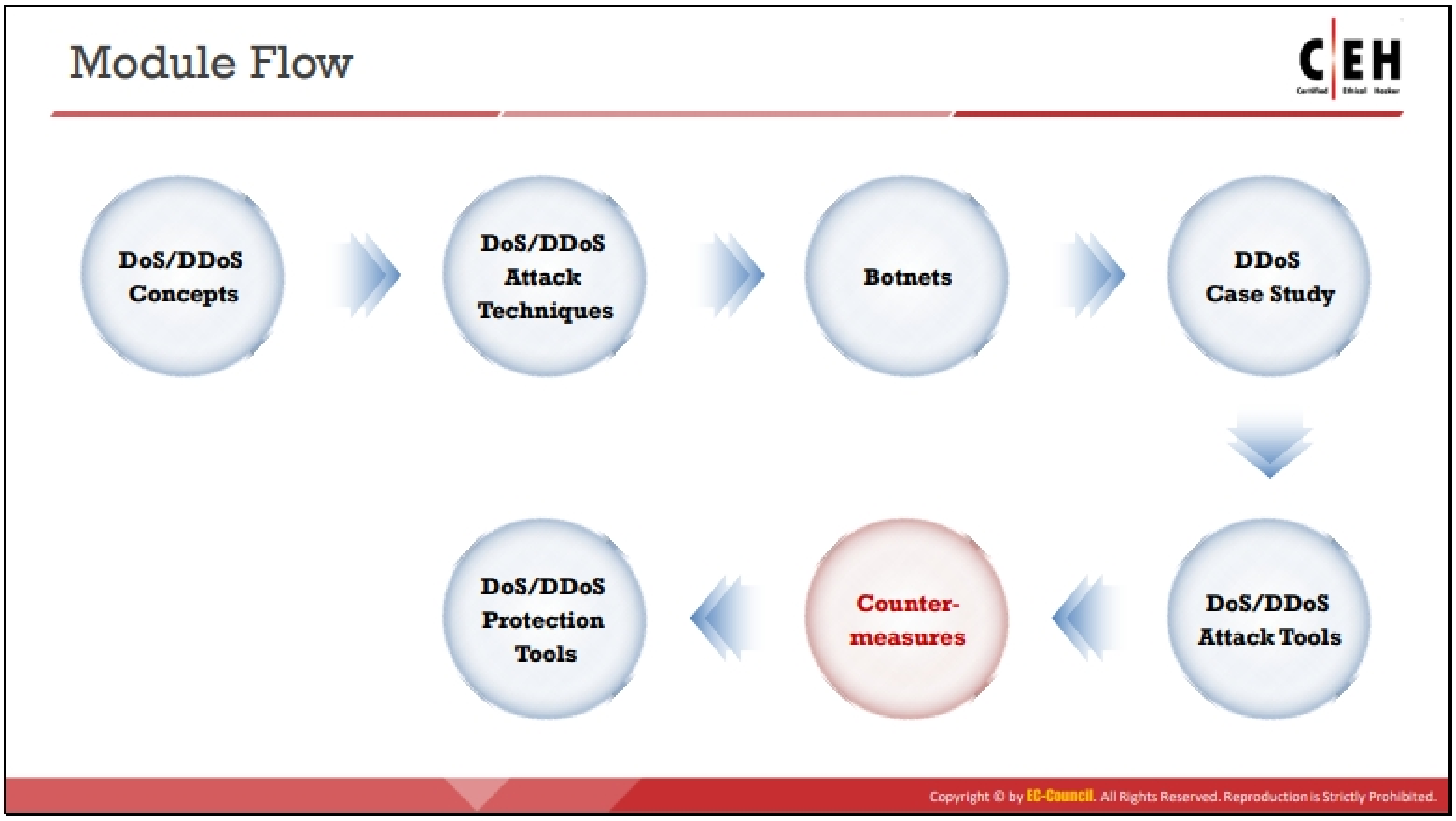
  The Packets Generator app allows attackers to generate network traffic including TCP SYN, UDP, and ICMP ping traffic. It is effectively used for testing a firewall's filtering rules, an intrusion detection system's attack signatures, and a router's access-control lists (ACLs) by sending a series of generated packets towards a target.



Transmission Control Protocol
TCP Syn traffic
Local IP: 10.153.76.122
Target IP: 10.153.76.25
Target port : 1-65535 ☑ Random
Number of packets : 80
Number of times : 5

Disclaimer: This app is for educational purposes only. Use it at your own risk!

Figure 10.32: Screenshot of Packets Generator tool for mobile



Module Flow

DoS/DDoS Concepts

DoS/DDoS Attack Techniques

Botnets

DDoS Case Study

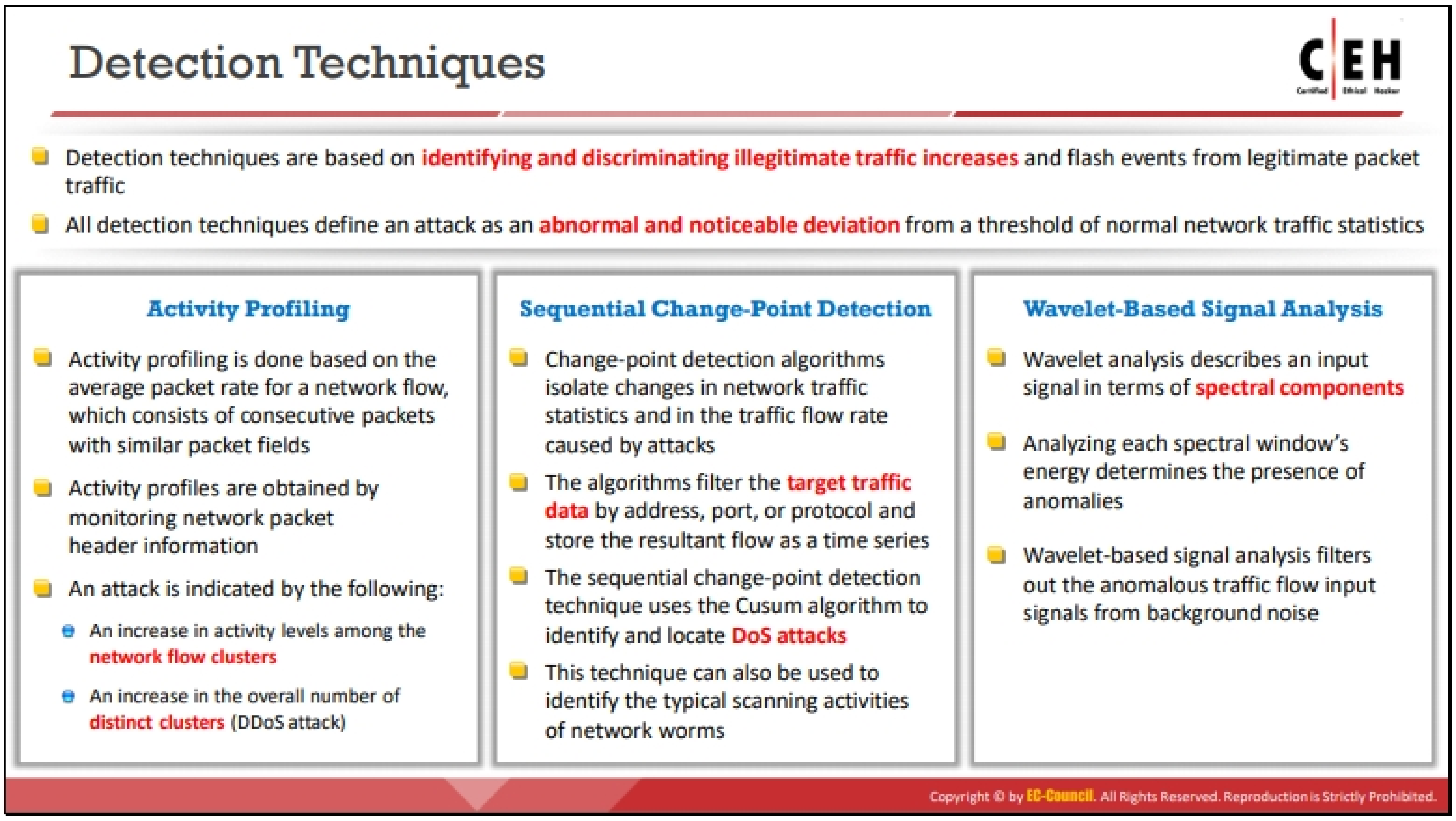
DoS/DDoS Attack Tools

Counter-measures

DoS/DDoS Protection Tools

Countermeasures

DoS/DDoS is one of the foremost security threats on the Internet; thus, there is a great necessity for solutions to mitigate these attacks. This section discusses detection methods, various preventive measures, and responses to DoS/DDoS attacks.



Detection Techniques

- Detection techniques are based on **identifying and discriminating illegitimate traffic increases** and flash events from legitimate packet traffic
- All detection techniques define an attack as an **abnormal and noticeable deviation** from a threshold of normal network traffic statistics

Activity Profiling

- Activity profiling is done based on the average packet rate for a network flow, which consists of consecutive packets with similar packet fields
- Activity profiles are obtained by monitoring network packet header information
- An attack is indicated by the following:
  - An increase in activity levels among the **network flow clusters**
  - An increase in the overall number of **distinct clusters** (DDoS attack)

Sequential Change-Point Detection

- Change-point detection algorithms isolate changes in network traffic statistics and in the traffic flow rate caused by attacks
- The algorithms filter the **target traffic data** by address, port, or protocol and store the resultant flow as a time series
- The sequential change-point detection technique uses the Cusum algorithm to identify and locate **DoS attacks**
- This technique can also be used to identify the typical scanning activities of network worms

Wavelet-Based Signal Analysis

- Wavelet analysis describes an input signal in terms of **spectral components**
- Analyzing each spectral window's energy determines the presence of anomalies
- Wavelet-based signal analysis filters out the anomalous traffic flow input signals from background noise

Detection Techniques

Early detection techniques help prevent DoS/DDoS attacks. Detecting a DoS/DDoS attack is a tricky task. A DoS/DDoS attack traffic detector needs to distinguish between a genuine and a bogus data packet, which is not always possible. Therefore, the techniques employed for this purpose are not perfect. There is always a chance of confusion between traffic generated by a legitimate network user and traffic generated by a DoS/DDoS attack. Detection techniques are based on the identification and discrimination of an illegitimate traffic increase and flash events from legitimate packet traffic.

One problem in filtering bogus traffic from legitimate traffic is the volume of traffic. It is impossible to scan each data packet to ensure security from a DoS/DDoS attack.

All the detection techniques used today define an attack as an abnormal and noticeable deviation in network traffic statistics and characteristics. These techniques involve the statistical analysis of deviations to categorize malicious and genuine traffic.

The following are the three types of detection techniques:

- **Activity Profiling**

  Activity profiling is performed based on the average packet rate for network flow, which consists of consecutive packets with similar packet header information. The packet header information includes the IP addresses of the destination and sender, ports, and transport protocols used. An attack is indicated by

  - An increase in activity levels among the network flow clusters
  - An increase in the overall number of distinct clusters (DDoS attack)

For a higher average packet rate or activity level of a flow, the time between consecutive matching packets is lower. Randomness in the average packet rate or activity level can indicate suspicious activity. The entropy calculation method measures randomness in activity levels. If a network is under attack, the entropy of network activity levels increases.

One of the major hurdles in the activity profiling method is the huge volume of traffic. This problem can be overcome by clustering packet flows with similar characteristics. Because DoS attacks generate a large number of data packets that are very similar, an increase in the average packet rate or an increase in the diversity of packets could indicate a DoS attack.

- **Sequential Change-Point Detection**

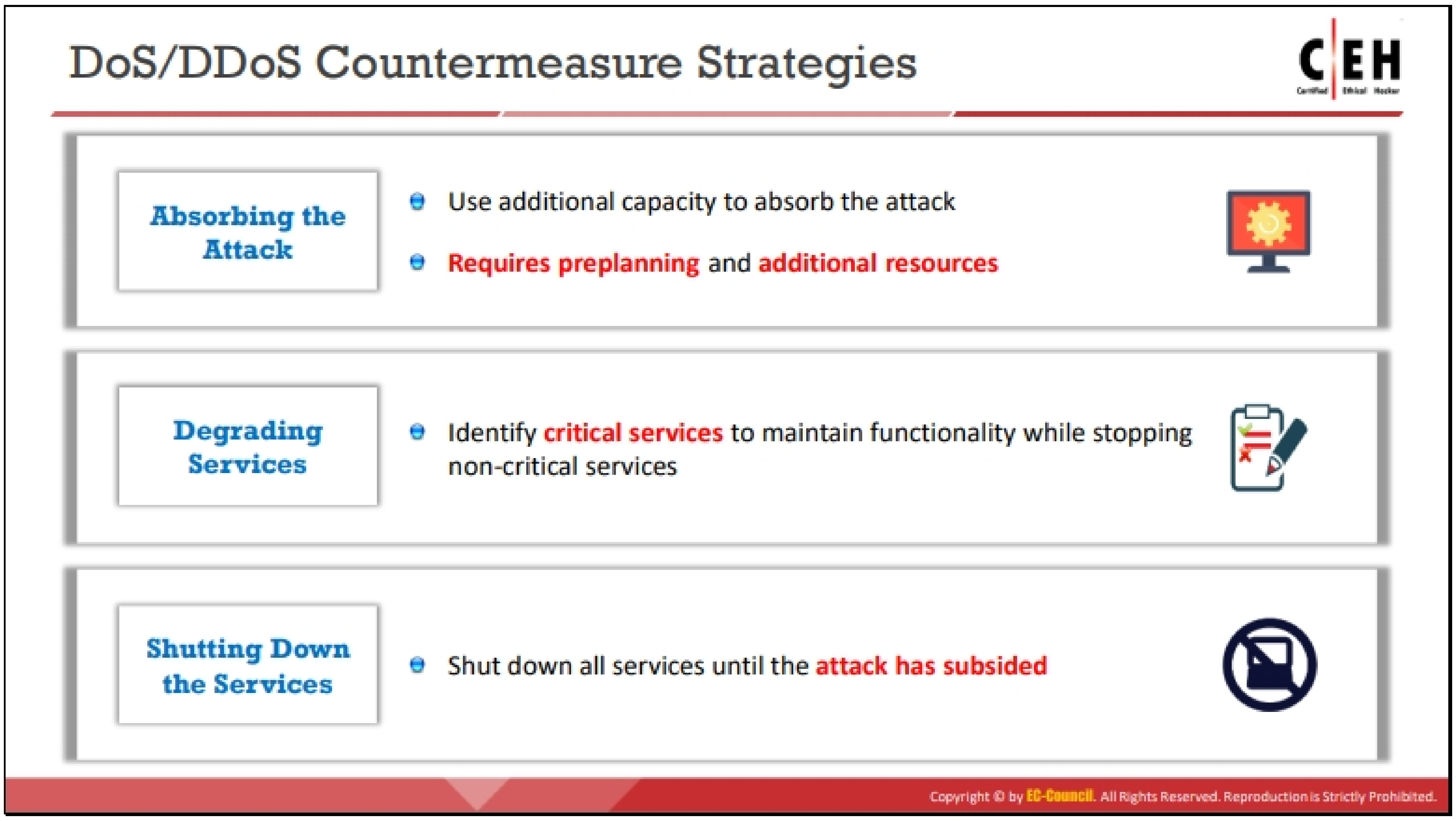
  In the sequential change-point detection technique, network traffic is filtered by IP addresses, targeted port numbers, and communication protocols used, and the traffic flow data are stored in a graph that shows the traffic flow rate versus time. Change-point detection algorithms isolate changes in network traffic statistics and in traffic flow rate caused by attacks. If there is a drastic change in traffic flow rate, a DoS attack may be occurring.

  This technique uses the cumulative sum (CUSUM) algorithm to identify and locate DoS attacks. The algorithm calculates deviations in the actual versus expected local average in the traffic time series. The sequential change-point detection technique identifies the typical scanning activities of network worms.

- **Wavelet-Based Signal Analysis**

  The wavelet analysis technique analyzes network traffic in terms of spectral components. It divides incoming signals into various frequencies and analyzes different frequency components separately. Analyzing each spectral window's energy reveals the presence of anomalies. These techniques check frequency components present at a specific time and provide a description of those components. The presence of an unfamiliar frequency indicates suspicious network activity.

  A network signal consists of a time-localized data packet flow signal and background noise. Wavelet-based signal analysis filters out the input signals of anomalous traffic flow from background noise. Normal network traffic is generally low-frequency traffic. During an attack, the high-frequency components of a signal increase.



DoS/DDoS Countermeasure Strategies

Absorbing the Attack

- Use additional capacity to absorb the attack
- Requires preplanning and additional resources

Degrading Services

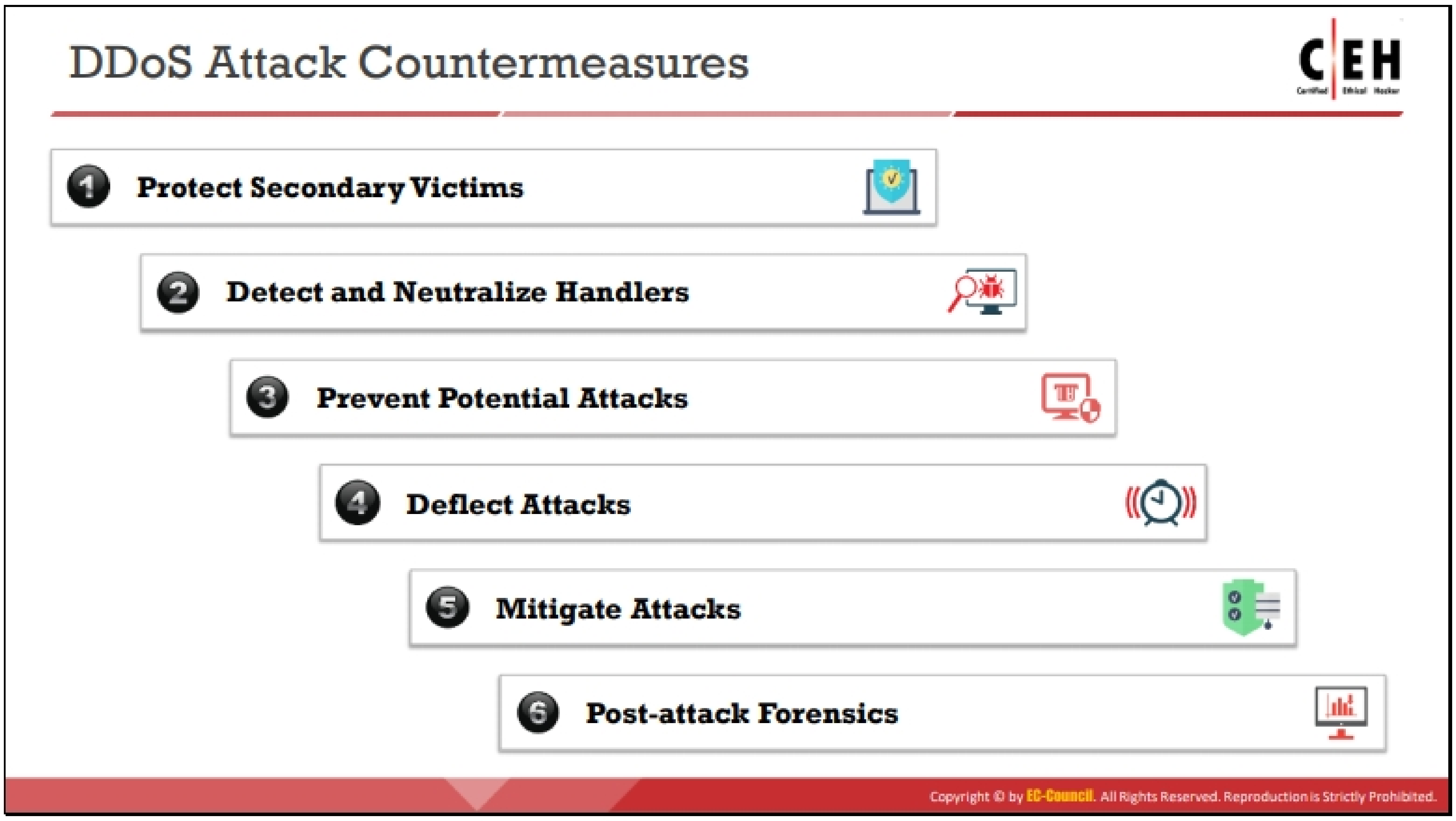
- Identify critical services to maintain functionality while stopping non-critical services

Shutting Down the Services

- Shut down all services until the attack has subsided

DoS/DDoS Countermeasure Strategies

- **Absorbing the Attack**: In this strategy, additional capacity is used to absorb an attack, which requires preplanning. It also requires additional resources. One disadvantage associated with this strategy is the cost of additional resources, which is incurred even when no attacks are underway.
- **Degrading Services**: If it is not possible to keep all services functioning during an attack, it is a good idea to keep at least the critical services functional. For this, the critical services are first identified, following which the network, systems, and application designs are customized to cut down the noncritical services. This strategy may help keep the critical services functional.
- **Shutting Down Services**: In this strategy, all services are shut down until an attack has subsided. Though it may not be the ideal choice, it may be a reasonable response in some cases.



DDoS Attack Countermeasures

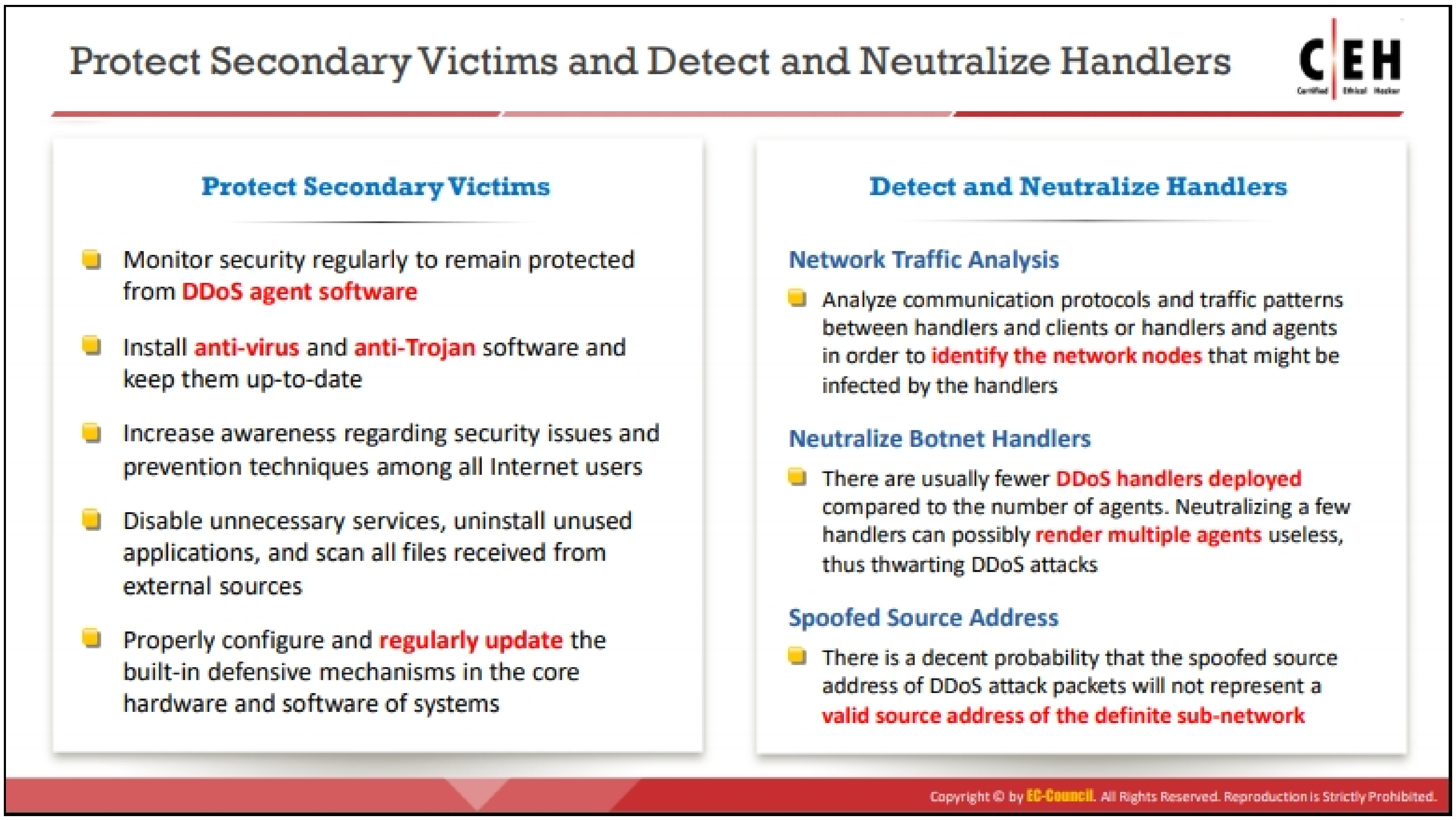
1. **Protect Secondary Victims**
2. **Detect and Neutralize Handlers**
3. **Prevent Potential Attacks**
4. **Deflect Attacks**
5. **Mitigate Attacks**
6. **Post-attack Forensics**

DDoS Attack Countermeasures

Many solutions have been proposed for mitigating the effects of a DDoS attack. However, no single complete solution exists that can protect all known forms of DDoS attacks. Moreover, attackers continually devise new methods to perform DDoS attacks to bypass the security solutions employed.

The following are examples for DDoS attack countermeasures:

- Protect secondary victims
- Neutralize handlers
- Prevent potential attacks
- Deflect attacks
- Mitigate attacks
- Post-attack forensics



Protect Secondary Victims and Detect and Neutralize Handlers

Protect Secondary Victims

- Monitor security regularly to remain protected from **DDoS agent software**
- Install **anti-virus** and **anti-Trojan** software and keep them up-to-date
- Increase awareness regarding security issues and prevention techniques among all Internet users
- Disable unnecessary services, uninstall unused applications, and scan all files received from external sources
- Properly configure and **regularly update** the built-in defensive mechanisms in the core hardware and software of systems

Detect and Neutralize Handlers

Network Traffic Analysis

- Analyze communication protocols and traffic patterns between handlers and clients or handlers and agents in order to **identify the network nodes** that might be infected by the handlers

Neutralize Botnet Handlers

- There are usually fewer **DDoS handlers deployed** compared to the number of agents. Neutralizing a few handlers can possibly **render multiple agents** useless, thus thwarting DDoS attacks

Spoofed Source Address

- There is a decent probability that the spoofed source address of DDoS attack packets will not represent a **valid source address of the definite sub-network**

Protect Secondary Victims

- **Individual Users**

  The best method to prevent DDoS attacks is for secondary victim systems to prevent themselves from taking part in the attack. This demands intensified security awareness and prevention techniques. Secondary victims must monitor their security regularly to remain protected from DDoS agent software. It must be ensured that the system does not install any DDoS agent program; further, DDoS agent traffic must not be transferred into the network.

  Antivirus and anti-Trojan software must be installed and updated regularly, as well as software patches to fix known vulnerabilities. Moreover, awareness of security issues and prevention techniques must be increased among all Internet users. It is important to disable unnecessary services, uninstall unused applications, and scan all files received from external sources. Because these tasks may appear daunting to the average web user, the core hardware and software of computing systems come with integrated mechanisms that defend against malicious code insertion. Therefore, the built-in defensive mechanisms in the core hardware and software of the systems must be properly configured and regularly updated to avoid DDoS attacks. Employing the above countermeasures will leave attackers with no DDoS attack network through which they can launch DDoS attacks.

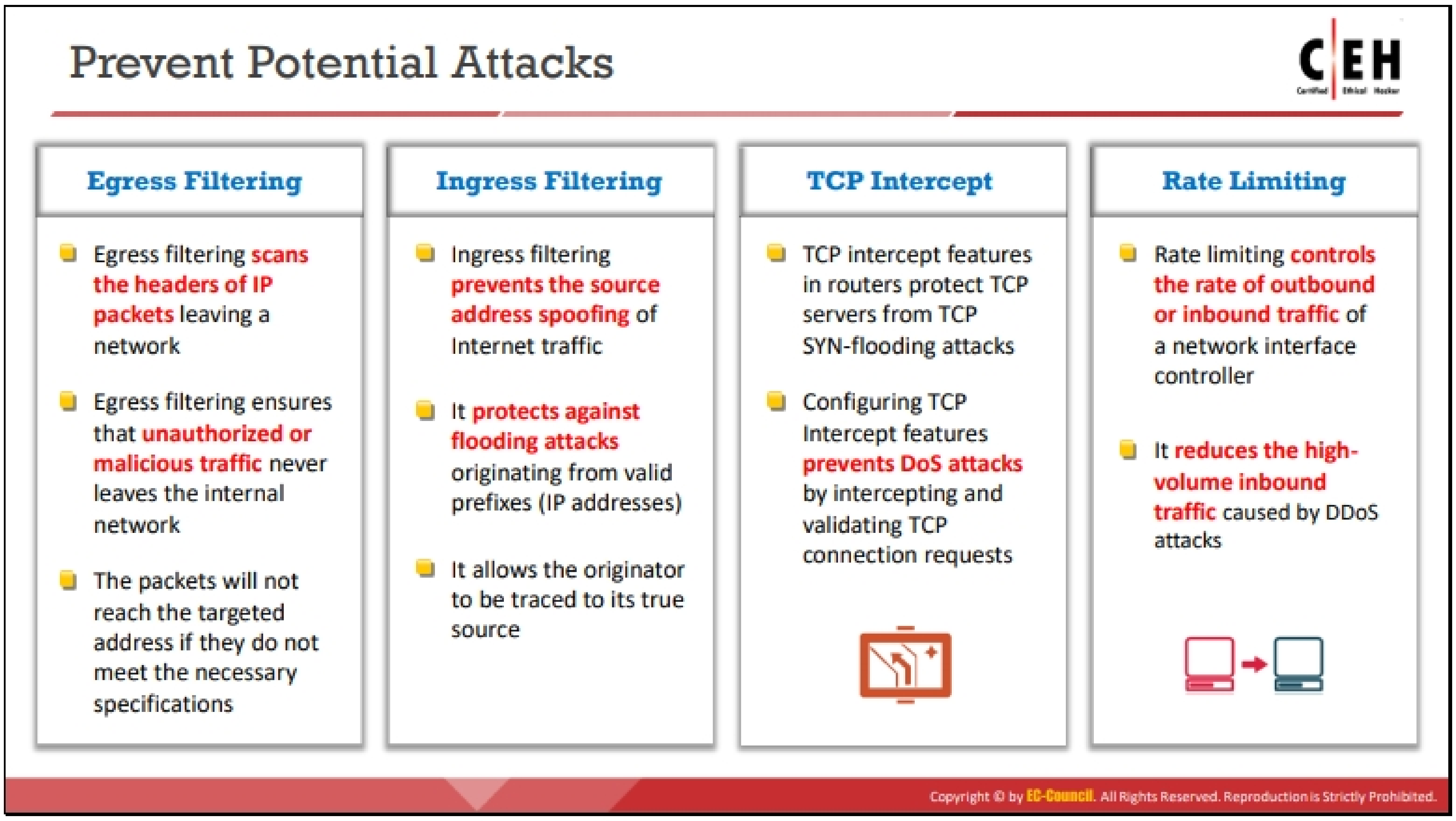
- **Network Service Providers**

  Service providers and network administrators can adopt dynamic pricing for their network usage to charge potential secondary victims for accessing the Internet and thereby encourage them to become more active in preventing themselves from becoming a part of a DDoS attack.

Detect and Neutralize Handlers

An important method used to stop DDoS attacks is to detect and neutralize handlers. This can be achieved by network traffic analysis, neutralizing botnet handlers, and identifying spoofed source addresses. In the agent-handler DDoS attack-tool arsenal, the handler works as an intermediary for the attacker to initiate attacks. Analyzing communication protocols and traffic patterns between handlers and clients or handlers and agents can reveal the network nodes infected by the handlers. Discovering the handlers in the network and disabling them can be a quick method of disrupting the DDoS attack network. Because the number of DDoS handlers deployed in the network is much less than the number of agents, neutralizing a few handlers can possibly render multiple agents useless, thereby thwarting DDoS attacks.

Furthermore, there is a reasonable probability that the spoofed source address of DDoS attack packets will not represent a valid source address of the definite sub-network. Identifying spoofed source addresses will prevent DDoS attacks with thorough comprehension of communication protocols and traffic among handlers, clients, and agents.



Prevent Potential Attacks

Egress Filtering

- Egress filtering scans the headers of IP packets leaving a network
- Egress filtering ensures that unauthorized or malicious traffic never leaves the internal network
- The packets will not reach the targeted address if they do not meet the necessary specifications

Ingress Filtering

- Ingress filtering prevents the source address spoofing of Internet traffic
- It protects against flooding attacks originating from valid prefixes (IP addresses)
- It allows the originator to be traced to its true source

TCP Intercept

- TCP intercept features in routers protect TCP servers from TCP SYN-flooding attacks
- Configuring TCP Intercept features prevents DoS attacks by intercepting and validating TCP connection requests

Rate Limiting

- Rate limiting controls the rate of outbound or inbound traffic of a network interface controller
- It reduces the high-volume inbound traffic caused by DDoS attacks

Prevent Potential Attacks

- **Egress Filtering**

  Egress filtering scans the headers of IP packets leaving a network. If the packets meet specifications, they can be routed out of the sub-network from which they originated. On the other hand, the packets do not reach the targeted address if they fail to meet the necessary specifications. Egress filtering ensures that unauthorized or malicious traffic never leaves the internal network.

  DDoS attacks generate spoofed IP addresses. Establishing protocols to require any legitimate packet that leaves a company's network to have a source address in which the network portion matches the internal network can help mitigate attacks. A properly developed firewall for the sub-network can filter out many DDoS packets with spoofed IP source addresses.

  If a web server is vulnerable to a zero-day attack known only to the underground hacker community, a server can be vulnerable even after applying all available patches. However, if the user enables egress filtering, they can save the integrity of a system by keeping the server from establishing a connection back to the attacker. This would also limit the effectiveness of many payloads used in common exploits. Outbound exposure can be restricted to the required traffic, thereby limiting the attacker's ability to connect to other systems and gain access to tools that can enable further access into the network.

- **Ingress Filtering**

  Ingress filtering is a packet filtering technique used by many Internet Service Providers (ISPs) to prevent the source address spoofing of Internet traffic. Thus, ingress filtering can indirectly combat several types of net abuse by making Internet traffic traceable to its true source. It protects against flooding attacks that originate from valid prefixes (IP addresses) and enables the originator to be traced to its true source.

- **TCP Intercept**

  TCP intercept is a traffic-filtering feature in routers to protect TCP servers from a TCP SYN-flooding attack, which is a kind of DoS attack. In a SYN-flooding attack, the attacker sends a huge volume of requests to connect with unreachable return addresses. As the addresses are not reachable, the connections cannot be established and remain unresolved. This huge volume of unresolved open connections overwhelms the server and may cause it to deny service even to valid requests. Consequently, legitimate users may not be able to connect to a website, access email, use an FTP service, and so on.

  In the TCP intercept mode, a router intercepts the SYN packets sent by clients to a server and matches them with an extended access list. If a match is obtained, then on behalf of the destination server, the intercept software establishes a connection with the client. Similarly, the intercept software also establishes a connection with the destination server on behalf of the client. Once the two half connections are established, the intercept software combines them transparently. Thus, the TCP intercept software prevents fake connection attempts from reaching the server by acting as a mediator between the server and client throughout the connection.

- **Rate limiting**

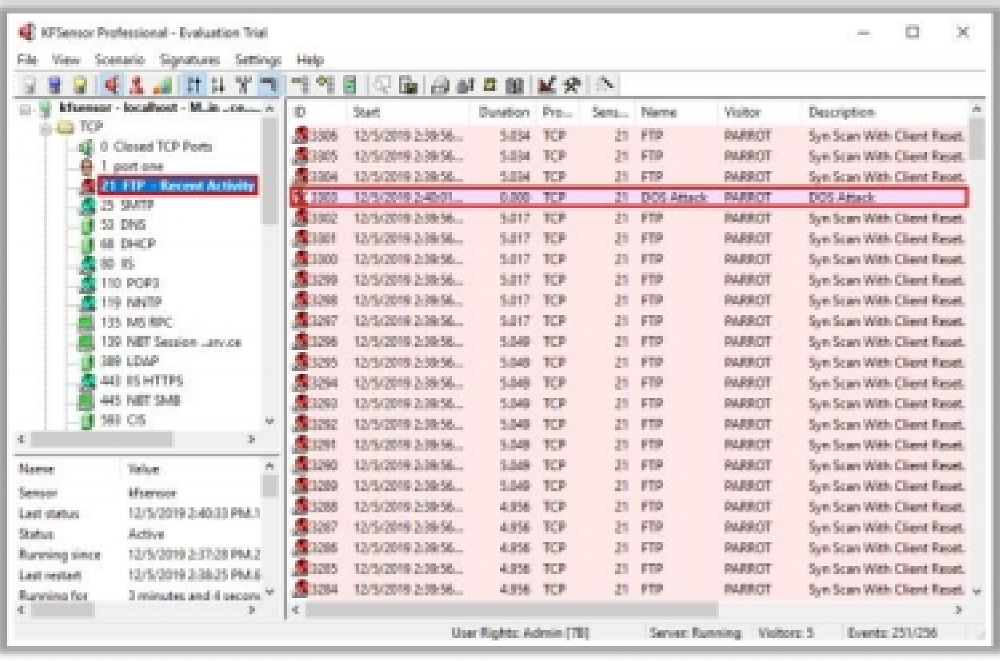
  Rate limiting is a technique used to control the rate of outbound or inbound traffic of a network interface controller. This technique effectively reduces the high volume of inbound traffic that causes a DDoS attack. It is especially important to employ this technique in hardware appliances, in which the technique is configured to limit the rate of requests on layers 4 and 5 of the Open Systems Interconnection (OSI) model.

Deflect Attacks

- Systems that are set up with limited security, also known as **Honeypots**, act as an enticement for an attacker
- Honeypots serve as a means of **gaining information** about attackers, **attack techniques**, and tools by storing a record of the system activities
- The defense-in-depth approach is used with IPSes at different network points to divert **suspicious DoS traffic** to several honeypots

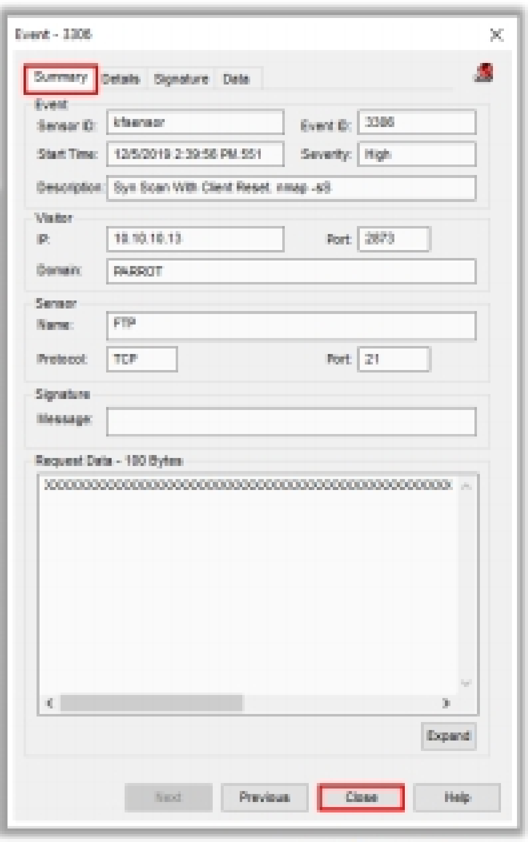
KFSensor

KFSensor acts as a honeypot, designed to attract and detect hackers and worms by simulating **vulnerable system services** and Trojans



```
ID    Start                Duration  Pro...  Sens...  Name        Visitor  Description
3306  12/5/2019 2:39:56... 5.034     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3305  12/5/2019 2:39:56... 5.034     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3304  12/5/2019 2:39:56... 5.034     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3303  12/5/2019 2:40:01... 0.000     TCP     21       DOS Attack  PARROT   DOS Attack
3302  12/5/2019 2:39:56... 5.017     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3301  12/5/2019 2:39:56... 5.017     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3300  12/5/2019 2:39:56... 5.017     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3299  12/5/2019 2:39:56... 5.017     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3298  12/5/2019 2:39:56... 5.017     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3297  12/5/2019 2:39:56... 5.017     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3296  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3295  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3294  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3293  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3292  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3291  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3290  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3289  12/5/2019 2:39:56... 5.049     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3288  12/5/2019 2:39:56... 4.956     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3287  12/5/2019 2:39:56... 4.956     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3286  12/5/2019 2:39:56... 4.956     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3285  12/5/2019 2:39:56... 4.956     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
3284  12/5/2019 2:39:56... 4.956     TCP     21       FTP         PARROT   Syn Scan With Client Reset.
```

```
Name            Value
Sensor          kfsensor
Last status     12/5/2019 2:40:33 PM.1
Status          Active
Running since   12/5/2019 2:37:28 PM.2
Last restart    12/5/2019 2:38:25 PM.6
Running for     3 minutes and 4 secon...
```



Event
Sensor ID: kfsensor
Event ID: 3306
Start Time: 12/5/2019 2:39:56 PM.551
Severity: High
Description: Syn Scan With Client Reset. nmap -sS

Visitor
IP: 10.10.10.13
Port: 2873
Domain: PARROT

Sensor
Name: FTP
Protocol: TCP
Port: 21

Signature
Message:

Request Data - 100 Bytes
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

http://www.keyfocus.net

Deflect Attacks

Systems set up with limited security, also known as honeypots, act as enticement for an attacker. Recent research reveals that a honeypot can imitate all aspects of a network, including its web servers, mail servers, and clients. Honeypots are intentionally set up with low security to gain the attention of DDoS attackers and serve as a means for gaining information about attackers, attack techniques, and tools by storing a record of the system activities. DDoS attackers attracted by a honeypot install handlers or agent code within the honeypot. This avoids compromising systems that are more sensitive. Honeypots not only protect the actual system from attackers but also keep track of details on the attackers' activities by recording the activity information. Consequently, the honeypot owner can keep a record of the handler and/or agent activity. Users can employ this knowledge to defend against any future DDoS installation attacks. A defense-in-depth approach with Internet Protocol Security (IPSec) can be used at different network points to divert suspicious DoS traffic to several honeypots.

There are two different types of honeypots:

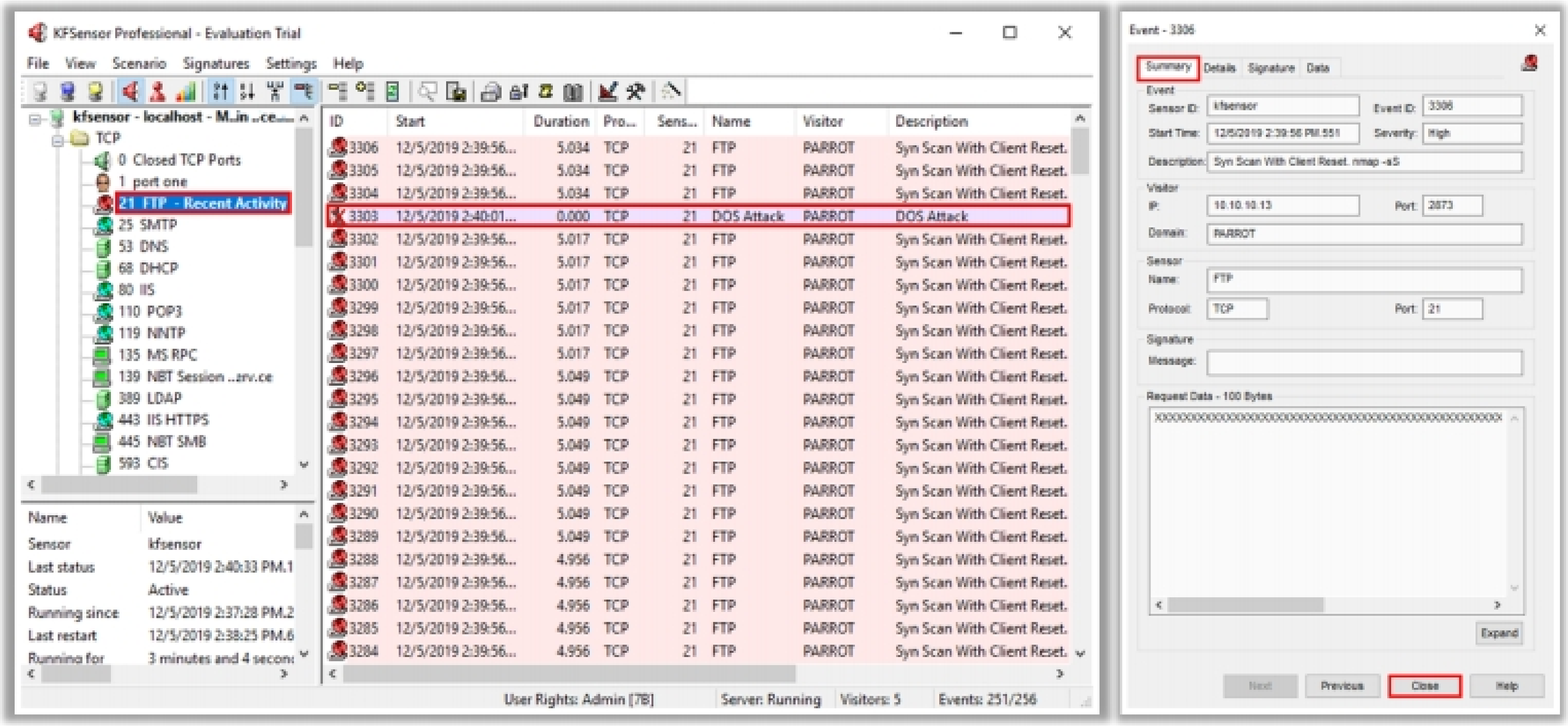
- Low-interaction honeypots
- High-interaction honeypots

An example for high-interaction honeypots is a honeynet. Honeynets form the security infrastructure; in other words, they simulate the complete layout of a network of computers but are originally intended for "capturing" attacks. The goal is to develop a network wherein all activities are controlled and tracked. This network contains potential victim decoys, and the network even has real computers running real applications.

- **KFSensor**

  Source: *http://www.keyfocus.net*

  KFSensor is a Windows-based honeypot intrusion detection system (IDS). It acts as a honeypot designed to attract and detect hackers and worms by simulating vulnerable system services and Trojans. By responding with an emulation of a real service, KFSensor can reveal the nature of an attack while maintaining total control and avoiding the risk of compromise. By acting as a decoy server, it can divert attacks from critical systems and provide a higher level of information than can be achieved using firewalls and a network IDS (NIDS) alone.


KFSensor Professional - Evaluation Trial
File
View
Scenario
Signatures
Settings
Help
kfsensor - localhost - M..in ..ce...
TCP
0 Closed TCP Ports
1 port one
21 FTP - Recent Activity
25 SMTP
53 DNS
68 DHCP
80 IIS
110 POP3
119 NNTP
135 MS RPC
139 NBT Session ..srv.ce
389 LDAP
443 IIS HTTPS
445 NBT SMB
593 CIS
User Rights: Admin [78]
Server: Running
Visitors: 5
Events: 251/256
Event - 3306
Summary
Details
Signature
Data
Next
Previous
Close
Help
Expand

Name	Value
Sensor	kfsensor
Last status	12/5/2019 2:40:33 PM.1
Status	Active
Running since	12/5/2019 2:37:28 PM.2
Last restart	12/5/2019 2:38:25 PM.6
Running for	3 minutes and 4 secon

ID	Start	Duration	Pro...	Sens...	Name	Visitor	Description
3306	12/5/2019 2:39:56...	5.034	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3305	12/5/2019 2:39:56...	5.034	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3304	12/5/2019 2:39:56...	5.034	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3303	12/5/2019 2:40:01...	0.000	TCP	21	DOS Attack	PARROT	DOS Attack
3302	12/5/2019 2:39:56...	5.017	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3301	12/5/2019 2:39:56...	5.017	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3300	12/5/2019 2:39:56...	5.017	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3299	12/5/2019 2:39:56...	5.017	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3298	12/5/2019 2:39:56...	5.017	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3297	12/5/2019 2:39:56...	5.017	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3296	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3295	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3294	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3293	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3292	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3291	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3290	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3289	12/5/2019 2:39:56...	5.049	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3288	12/5/2019 2:39:56...	4.956	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3287	12/5/2019 2:39:56...	4.956	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3286	12/5/2019 2:39:56...	4.956	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3285	12/5/2019 2:39:56...	4.956	TCP	21	FTP	PARROT	Syn Scan With Client Reset.
3284	12/5/2019 2:39:56...	4.956	TCP	21	FTP	PARROT	Syn Scan With Client Reset.

Event
Sensor ID: kfsensor
Event ID: 3306
Start Time: 12/5/2019 2:39:56 PM.551
Severity: High
Description: Syn Scan With Client Reset. nmap -sS

Visitor
IP: 10.10.10.13
Port: 2073
Domain: PARROT

Sensor
Name: FTP
Protocol: TCP
Port: 21

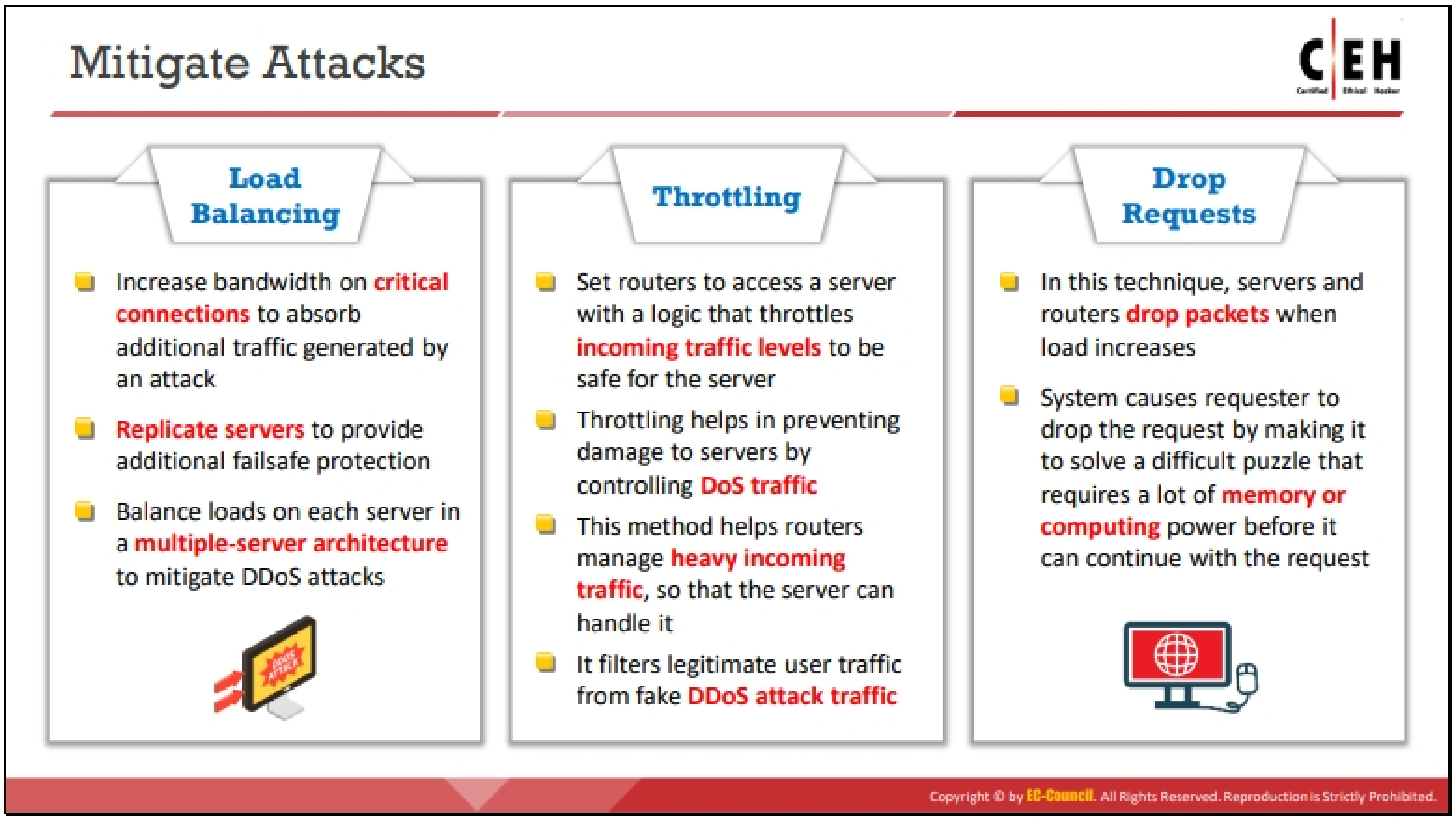
Signature
Message:

Request Data - 100 Bytes

Figure 10.33: Screenshot of KFSensor

The following are examples for additional DoS/DDoS countermeasure (honeypot) tools:

- SSHHiPot (*https://github.com*)
- Artillery (*https://github.com*)



Mitigate Attacks

Load Balancing

- Increase bandwidth on **critical connections** to absorb additional traffic generated by an attack
- **Replicate servers** to provide additional failsafe protection
- Balance loads on each server in a **multiple-server architecture** to mitigate DDoS attacks

Throttling

- Set routers to access a server with a logic that throttles **incoming traffic levels** to be safe for the server
- Throttling helps in preventing damage to servers by controlling **DoS traffic**
- This method helps routers manage **heavy incoming traffic**, so that the server can handle it
- It filters legitimate user traffic from fake **DDoS attack traffic**

Drop Requests

- In this technique, servers and routers **drop packets** when load increases
- System causes requester to drop the request by making it to solve a difficult puzzle that requires a lot of **memory or computing** power before it can continue with the request

Mitigate Attacks

- **Load Balancing**

  Bandwidth providers can increase bandwidth on critical connections in case of a DDoS attack to prevent their servers from shutting down. Using a replicated server model provides additional failsafe protection. Replicated servers help in better load management by balancing loads on each server in a multiple-server architecture; they also increase normal network performance and mitigate the effect of a DDoS attack.

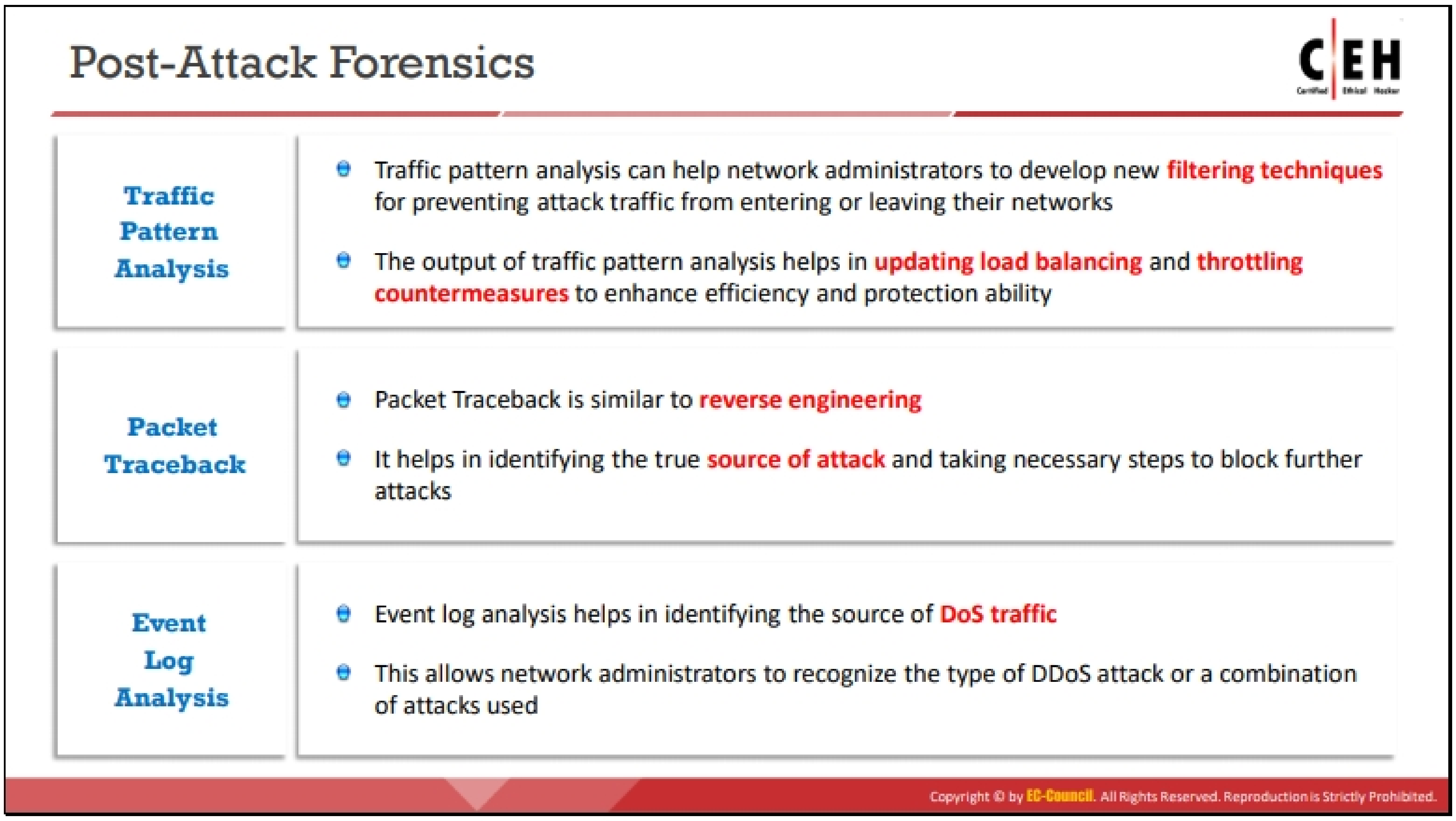
- **Throttling**

  Throttling entails the setting up of routers for server access with a logic to throttle incoming traffic levels that are safe for the server. "Min-max fair server-centric router" throttles (minimum and maximum throughput controls) help users prevent their servers from shutting down. Throttling helps in preventing damage to servers by controlling the DoS traffic. This method helps routers manage heavy incoming traffic so that the server can handle it. It also filters legitimate user traffic from fake DDoS attack traffic and can be extended to throttle DDoS attack traffic while allowing legitimate user traffic for better results.

  A major limitation of this method is that it may trigger false alarms. Occasionally, it may allow malicious traffic to pass through while dropping some legitimate traffic.

- **Drop Requests**

  Another method is to drop packets when the load increases. Usually, the router or server performs this task. However, before continuing with a request, the system induces the requester to drop the request by making them solve a difficult puzzle that requires a lot of memory or computing power. Consequently, users of zombie systems detect a performance degradation and could possibly be dissuaded from taking part in transferring DDoS attack traffic.



Post-Attack Forensics

Traffic Pattern Analysis	- Traffic pattern analysis can help network administrators to develop new **filtering techniques** for preventing attack traffic from entering or leaving their networks - The output of traffic pattern analysis helps in **updating load balancing** and **throttling countermeasures** to enhance efficiency and protection ability
Packet Traceback	- Packet Traceback is similar to **reverse engineering** - It helps in identifying the true **source of attack** and taking necessary steps to block further attacks
Event Log Analysis	- Event log analysis helps in identifying the source of **DoS traffic** - This allows network administrators to recognize the type of DDoS attack or a combination of attacks used

Post-Attack Forensics

- **Traffic Pattern Analysis**

  During a DDoS attack, the traffic pattern tool stores post-attack data, which users analyze to identify characteristics unique to the attacking traffic. These data are helpful in updating load balancing and throttling countermeasures to enhance their efficiency and protection ability. Moreover, DDoS attack traffic patterns can help network administrators develop new filtering techniques to prevent DDoS attack traffic from entering or leaving their networks. Analyzing DDoS traffic patterns can also help network administrators ensure that an attacker cannot use their servers as a DDoS platform to break into other sites.

- **Zombie Zapper Tool**

  When a company is unable to ensure the security of its servers and a DDoS attack starts, the network IDS notices the high volume of traffic, which indicates a potential problem. The targeted victim can run Zombie Zapper to stop packets from flooding the system.

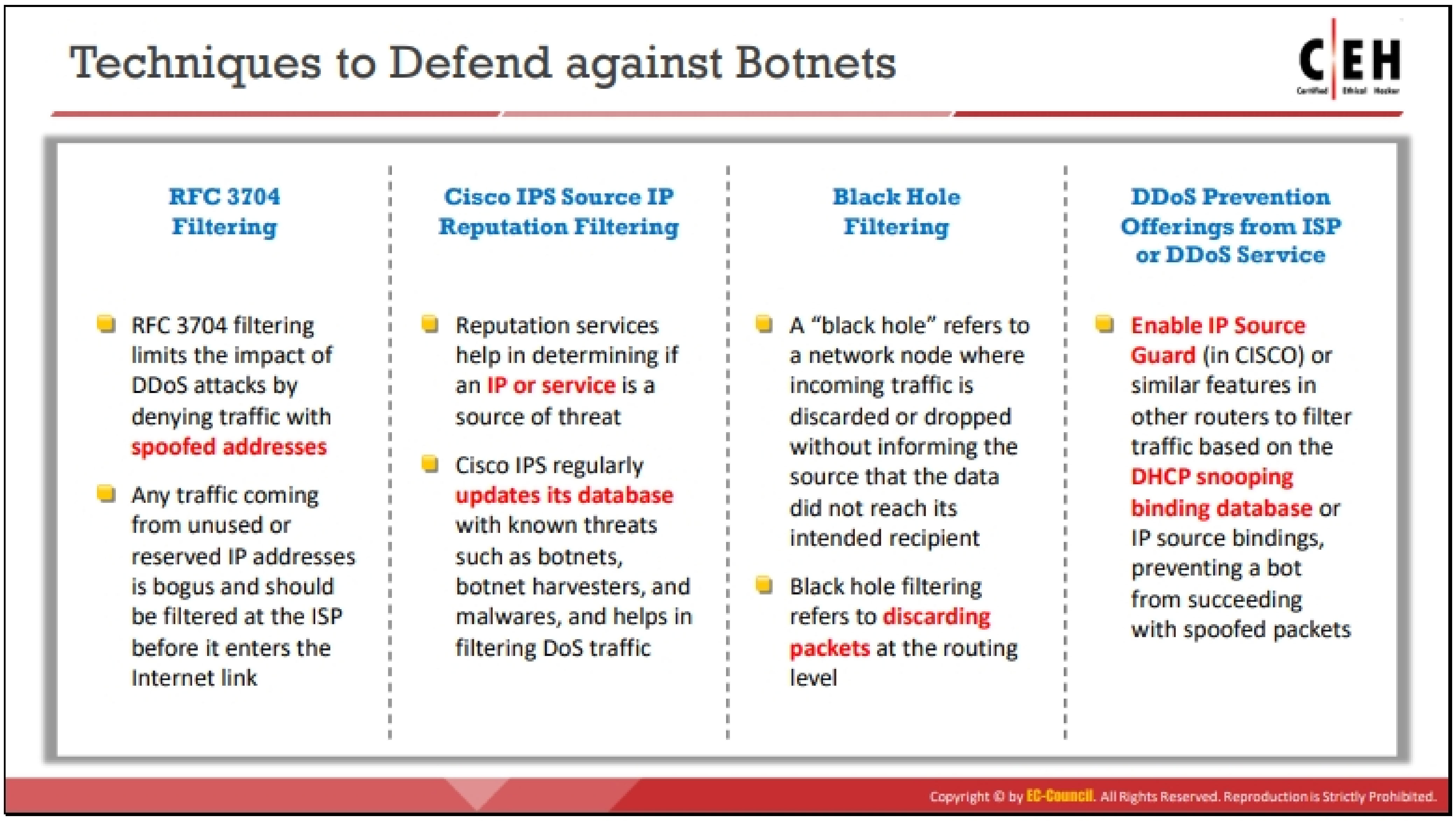
  There are two versions of Zombie Zapper: one runs on UNIX, while the other runs on Windows. Currently, this tool acts as a defense mechanism against Trinoo, Tribe Flood Network (TFN), Shaft, and Stacheldraht.

- **Packet Traceback**

  Packet traceback refers to tracing back attack traffic. It is similar to reverse engineering. In this method, the targeted victim works backward by tracing the packet to its source. Once the victim identifies the true source, they can take steps to block further attacks from that source by developing the necessary preventive techniques. In addition, packet traceback can assist in gaining knowledge of the various tools and techniques that an attacker uses. This information can help in developing and implementing different filtering techniques to block attacks.

- **Event Log Analysis**

  DDoS event logs assist in forensic investigation and the enforcement of laws, which are helpful when an attacker causes severe financial damage. Providers can use honeypots and other network security mechanisms such as firewalls, packet sniffers, and server logs to store all the events that occurred during the setup and execution of the attack. This allows network administrators to recognize the type of DDoS attack or the combination of attacks used. Routers, firewalls, and IDS logs can be analyzed to identify the source of the DoS traffic. Further, network administrators can attempt to trace back the attacker's IP address with the help of intermediary ISPs and law enforcement agencies.



Techniques to Defend against Botnets

RFC 3704 Filtering

- RFC 3704 filtering limits the impact of DDoS attacks by denying traffic with **spoofed addresses**
- Any traffic coming from unused or reserved IP addresses is bogus and should be filtered at the ISP before it enters the Internet link

Cisco IPS Source IP Reputation Filtering

- Reputation services help in determining if an **IP or service** is a source of threat
- Cisco IPS regularly **updates its database** with known threats such as botnets, botnet harvesters, and malwares, and helps in filtering DoS traffic

Black Hole Filtering

- A “black hole” refers to a network node where incoming traffic is discarded or dropped without informing the source that the data did not reach its intended recipient
- Black hole filtering refers to **discarding packets** at the routing level

DDoS Prevention Offerings from ISP or DDoS Service

- **Enable IP Source Guard** (in CISCO) or similar features in other routers to filter traffic based on the **DHCP snooping binding database** or IP source bindings, preventing a bot from succeeding with spoofed packets

Techniques to Defend against Botnets

There are four techniques to defend against botnets:

- **RFC 3704 Filtering**

  RFC 3704 is a basic access-control list (ACL) filter, which limits the impact of DDoS attacks by blocking traffic with spoofed addresses. This filter requires packets sourced from valid, allocated address space that is consistent with the topology and space allocation. A “bogon list” consists of all unused or reserved IP addresses that should not come from the Internet. If a packet is sourced from any of the IP addresses from the bogon list, then the packet is from a spoofed source IP, and the filter should drop it. System administrators should check whether the ISP performs RFC 3704 filtering in the cloud before traffic enters the system. Because the bogon list changes regularly, in case the ISP does not perform RFC 3704 filtering, the system administrator must manage their own bogon ACL rules or switch to another ISP.

- **Cisco IPS Source IP Reputation Filtering**

  Reputation services help in determining whether an IP or service is a source of threat. Cisco Global Correlation, a new security capability of Cisco IPS 7.0, uses immense security intelligence. The Cisco SensorBase Network contains information about all known threats on the Internet, such as botnets, malware outbreaks, dark nets, and botnet harvesters. The Cisco IPS makes use of this network to filter DoS traffic before it damages critical assets. To detect and prevent malicious activity even earlier, it incorporates global threat data into its system.

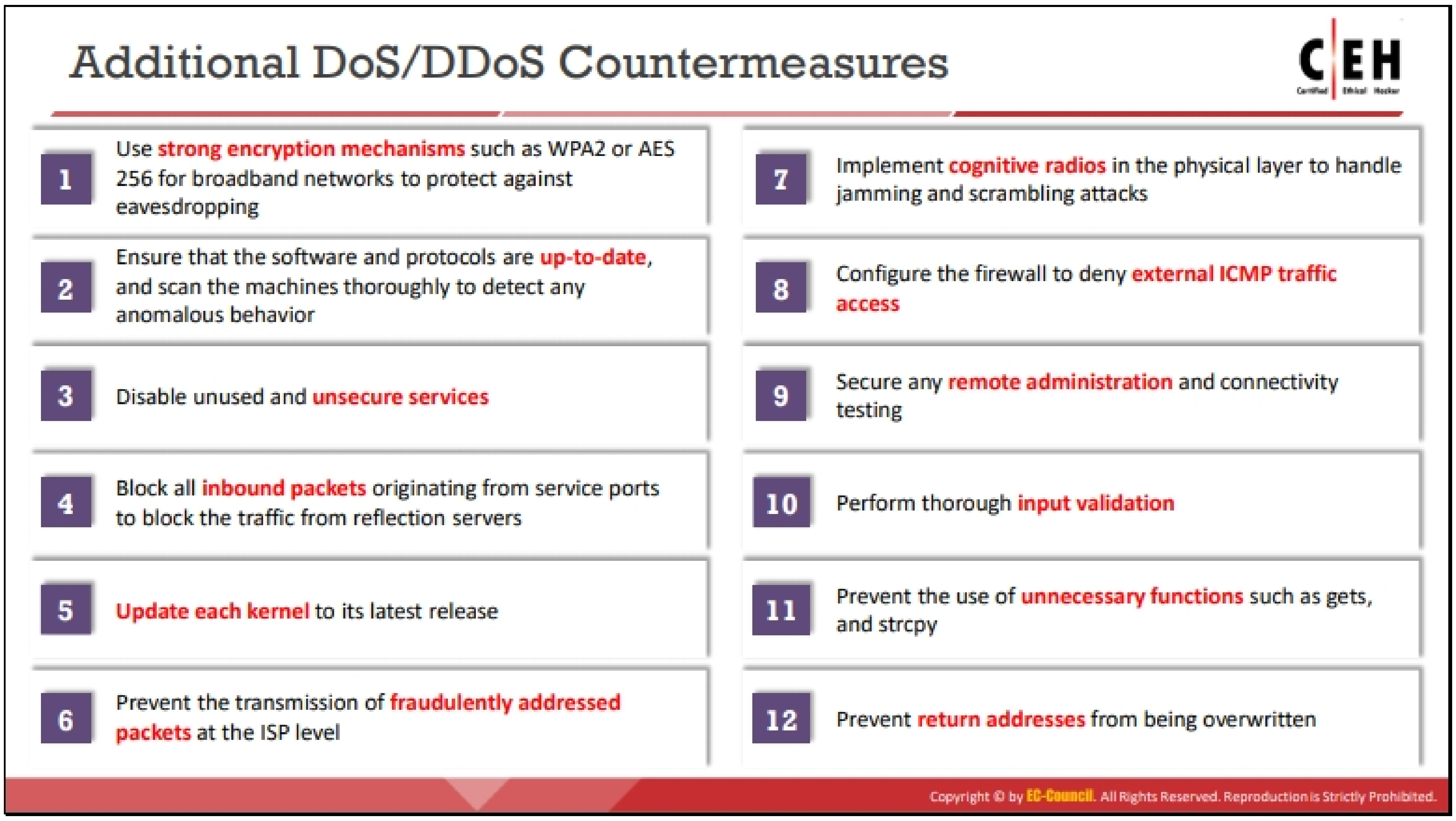
- **Black Hole Filtering**

  Black-hole filtering is a common technique to defend against botnets and, thus, to prevent DoS attacks. Black holes refer to network nodes wherein incoming traffic is discarded or dropped without informing the source that the data did not reach the intended recipient. Undesirable traffic can be dropped before it enters a protected network with a technique called remotely triggered black-hole (RTBH) filtering. As this is a remotely triggered process, this filtering must be performed in conjunction with the ISP. It uses Border Gateway Protocol (BGP) host routes to route traffic to the victim's servers to a "null0" next hop.

- **DDoS Prevention Offerings from ISP or DDoS Service**

  This method is effective in preventing IP spoofing at the ISP level. Here, the ISP scrubs/cleans traffic before allowing it to enter a user's Internet link. Because this service runs in the cloud, DDoS attacks do not saturate the Internet links. In addition, some third parties offer cloud DDoS prevention services.

  IP Source Guard (in CISCO) or similar features can be enabled in other routers to filter traffic based on the DHCP snooping binding database or IP source bindings, which prevent bots from sending spoofed packets.



Additional DoS/DDoS Countermeasures

1 Use **strong encryption mechanisms** such as WPA2 or AES 256 for broadband networks to protect against eavesdropping

2 Ensure that the software and protocols are **up-to-date**, and scan the machines thoroughly to detect any anomalous behavior

3 Disable unused and **unsecure services**

4 Block all **inbound packets** originating from service ports to block the traffic from reflection servers

5 **Update each kernel** to its latest release

6 Prevent the transmission of **fraudulently addressed packets** at the ISP level

7 Implement **cognitive radios** in the physical layer to handle jamming and scrambling attacks

8 Configure the firewall to deny **external ICMP traffic access**

9 Secure any **remote administration** and connectivity testing

10 Perform thorough **input validation**

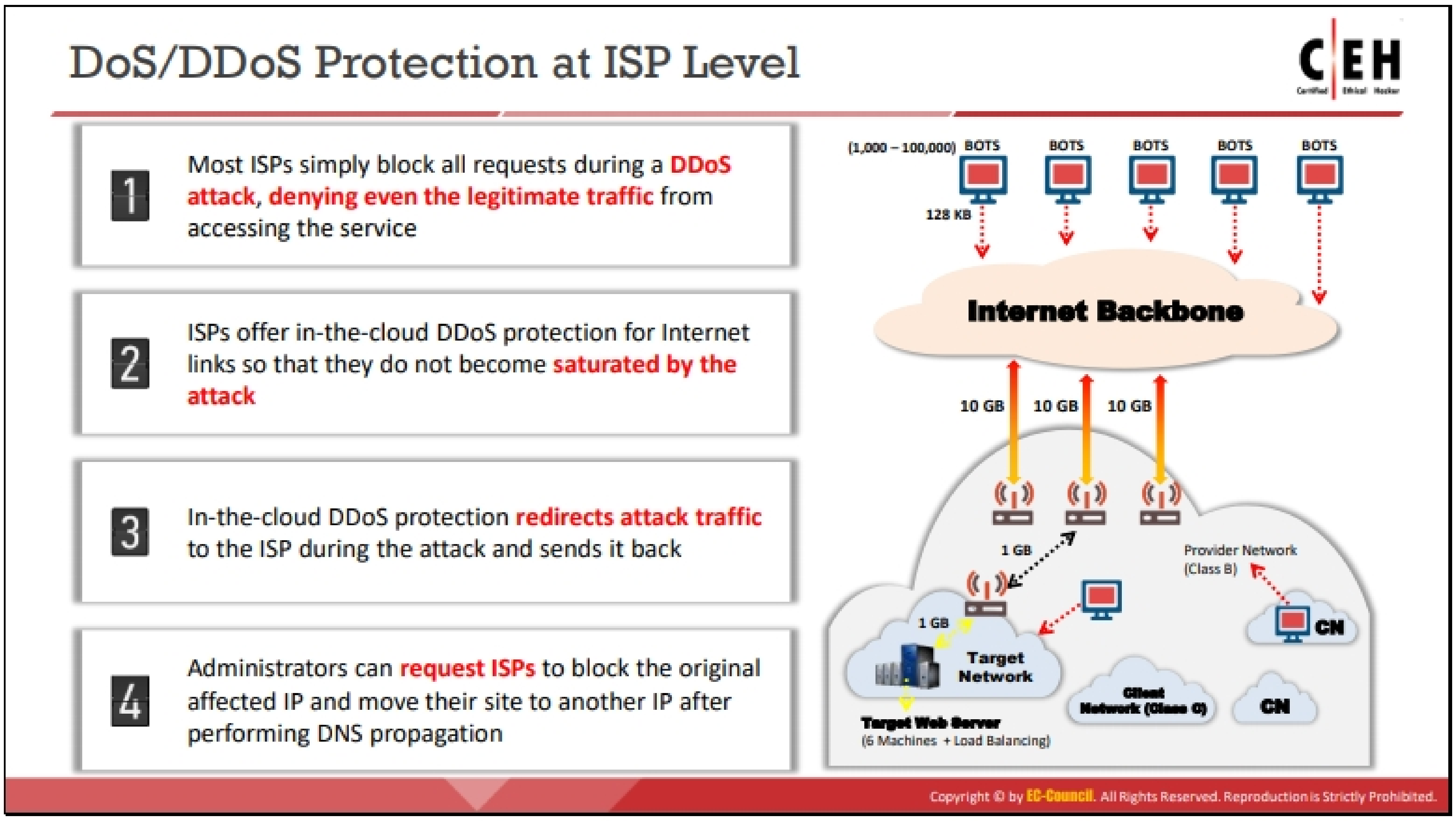
11 Prevent the use of **unnecessary functions** such as gets, and strcpy

12 Prevent **return addresses** from being overwritten

Additional DoS/DDoS Countermeasures

Implementing defensive mechanisms at proper places by following proper measures allows the heightening of organizational network security. The following is a list of countermeasures for combating DoS/DDoS attacks:

- Use strong encryption mechanisms such as WPA2 and AES 256 for broadband networks to defend against eavesdropping
- Ensure that the software and protocols are up-to-date and scan the machines thoroughly to detect any anomalous behavior
- Update the kernel to the latest release and disable unused and insecure services
- Block all inbound packets originating from the service ports to block the traffic from reflection servers
- Enable TCP SYN cookie protection
- Prevent the transmission of fraudulently addressed packets at the ISP level
- Implement cognitive radios in the physical layer to handle jamming and scrambling attacks
- Configure the firewall to deny external ICMP traffic access
- Secure remote administration and connectivity testing
- Perform thorough input validation
- Stop data processed by the attacker from being executed
- Prevent the use of unnecessary functions such as gets and strcpy
- Prevent the return addresses from being overwritten



DoS/DDoS Protection at ISP Level

1. Most ISPs simply block all requests during a **DDoS attack**, **denying even the legitimate traffic** from accessing the service
2. ISPs offer in-the-cloud DDoS protection for Internet links so that they do not become **saturated by the attack**
3. In-the-cloud DDoS protection **redirects attack traffic** to the ISP during the attack and sends it back
4. Administrators can **request ISPs** to block the original affected IP and move their site to another IP after performing DNS propagation

DoS/DDoS Protection at ISP Level

One of the best ways to defend against DoS attacks is to block them at the gateway. This task is performed by the contracted ISP. ISPs offer a “clean pipes” service-level agreement that provides an assured bandwidth of genuine traffic, rather than the total bandwidth of all traffic. Most ISPs simply block all requests during a DDoS attack, denying even legitimate traffic from accessing the service. If an ISP does not provide clean-pipes services, subscription services provided by many cloud service providers can be used. The subscription services serve as an intermediary, receive traffic destined for the network, filter it, and then pass on only trusted connections. Vendors such as Imperva and VeriSign offer services for cloud protection against DoS attacks.

ISPs offer in-the-cloud DDoS protection for Internet links to avoid saturation due to an attack. This type of protection redirects attack traffic to the ISP during an attack. Administrators can request ISPs to block the original affected IP and move their site to another IP after performing DNS propagation.

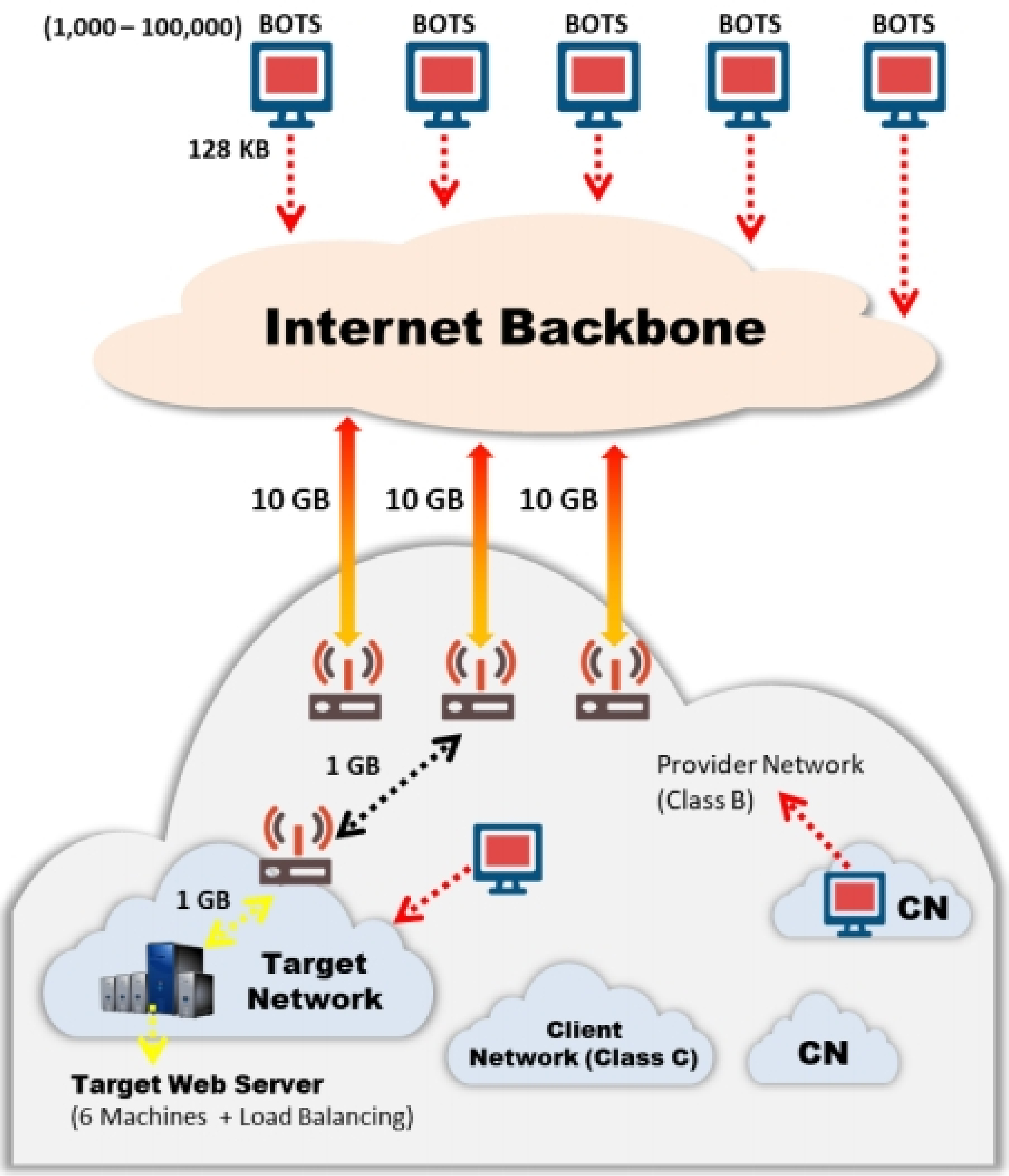
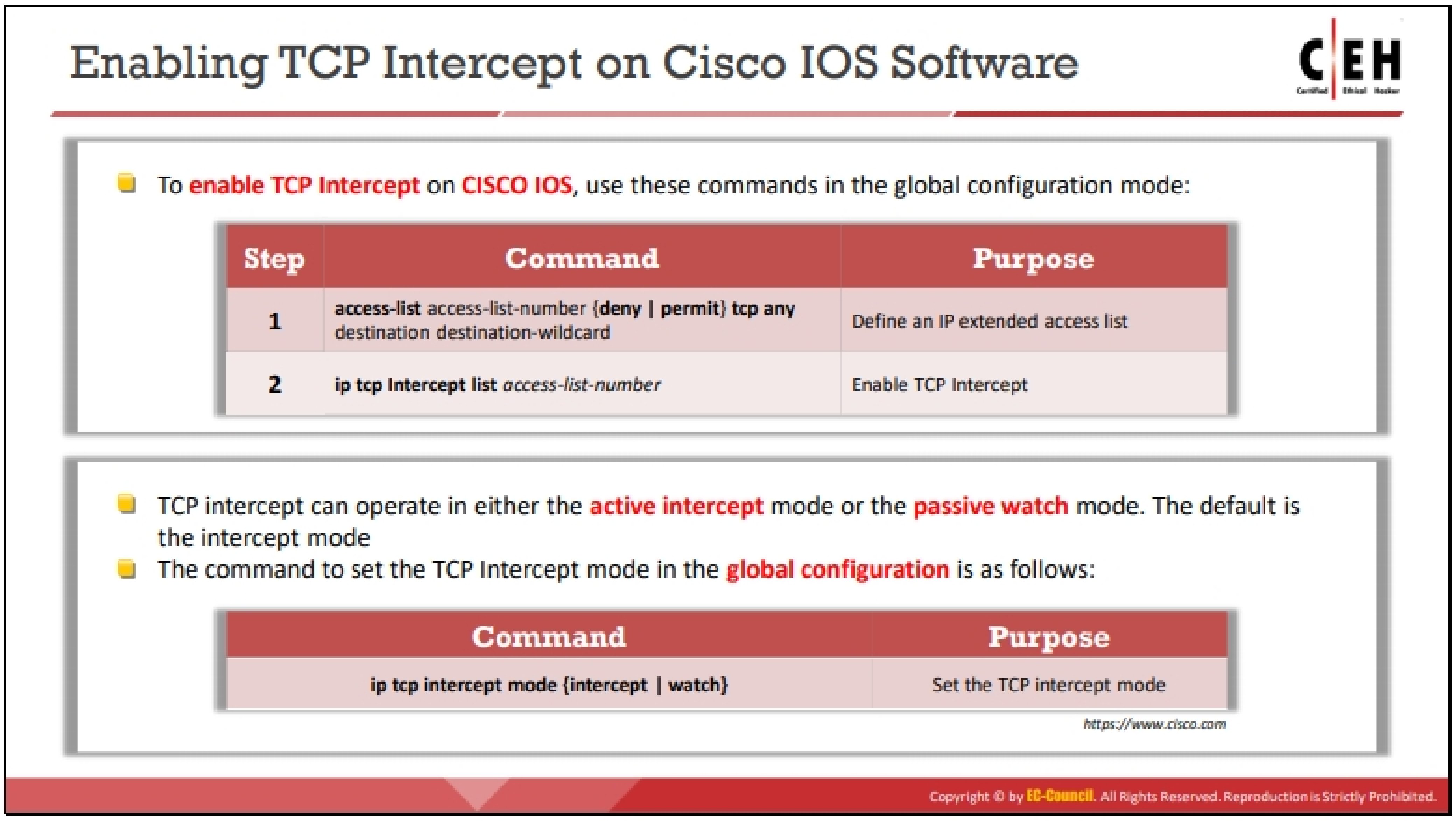

(1,000 – 100,000) BOTS
BOTS
BOTS
BOTS
BOTS
128 KB
Internet Backbone
10 GB
10 GB
10 GB
1 GB
Provider Network
(Class B)
1 GB
Target
Network
CN
Client
Network (Class C)
CN
Target Web Server
(6 Machines + Load Balancing)

Figure 10.34: DoS/DDoS protection at the ISP level



Enabling TCP Intercept on Cisco IOS Software

- To **enable TCP Intercept** on **CISCO IOS**, use these commands in the global configuration mode:

Step	Command	Purpose
1	**access-list** access-list-number {**deny | permit**} **tcp any** destination destination-wildcard	Define an IP extended access list
2	**ip tcp Intercept list** *access-list-number*	Enable TCP Intercept

- TCP intercept can operate in either the **active intercept** mode or the **passive watch** mode. The default is the intercept mode
- The command to set the TCP Intercept mode in the **global configuration** is as follows:

Command	Purpose
ip tcp intercept mode {intercept | watch}	Set the TCP intercept mode

https://www.cisco.com

Enabling TCP Intercept on Cisco IOS Software

Source: *https://www.cisco.com*

TCP intercept can be enabled by executing the commands given in the below table in the global configuration mode.

Step	Command	Purpose
1	`access-list access-list-number {deny | permit} tcp any destination destination-wildcard`	Defines an IP extended access list
2	`ip tcp intercept list access-list-number`	Enables TCP intercept

Table 10.1: Steps to enable TCP intercept on Cisco IOS

An access list achieves three purposes:

1. Interception of all requests
2. Interception of only requests originating from specific networks
3. Interception of only requests destined for specific servers

Typically, an access list defines the source as any source and the destination as specific networks or servers. As it is unimportant to know who to intercept packets from, the source addresses are not filtered. Rather, the destination server or network to be protected is identified. TCP intercept can operate in either the active intercept mode or passive watch mode. The default is the intercept mode.

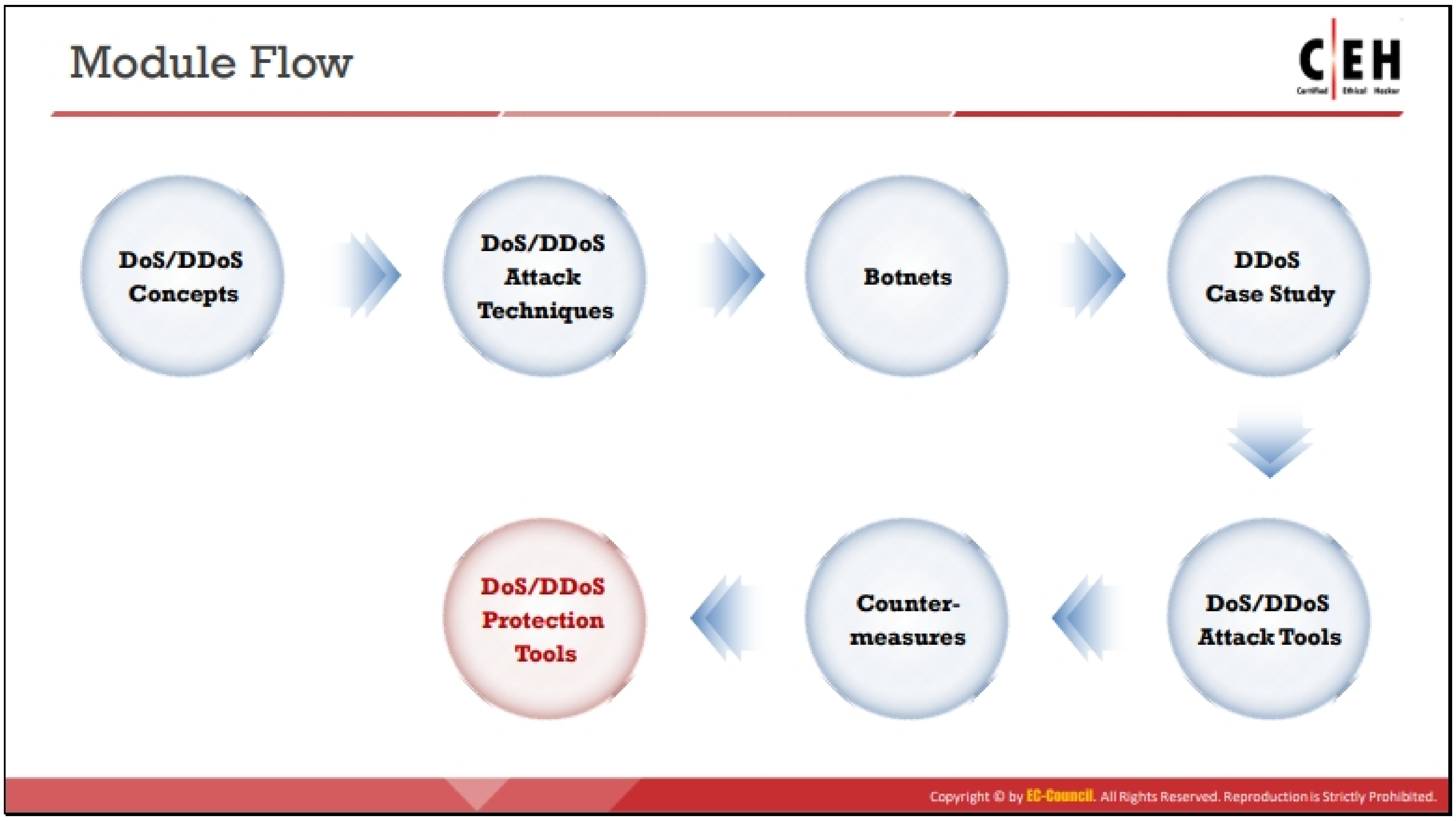
In the active intercept mode, the Cisco IOS software actively intercepts all inbound connection requests (SYN) and replies with a SYN-ACK on behalf of the server, following which it waits for an acknowledge (ACK) from the client. On receiving the ACK from the client, the server sends the original SYN, and the software makes a three-way handshake with the server. Once the three-way handshake is complete, the two half connections are linked.

In the passive watch mode, the user sends connection requests that pass through the server, but they need to wait until the connection is established. If connection requests fail to establish within 30 s, the software sends a reset request to the server to clear its state.

Below table presents the command to set the TCP intercept mode in the global configuration mode.

Command	**Purpose**
`ip tcp intercept mode {intercept | watch}`	Set the TCP intercept mode

Table 10.2: Command to set the TCP intercept mode in the global configuration mode

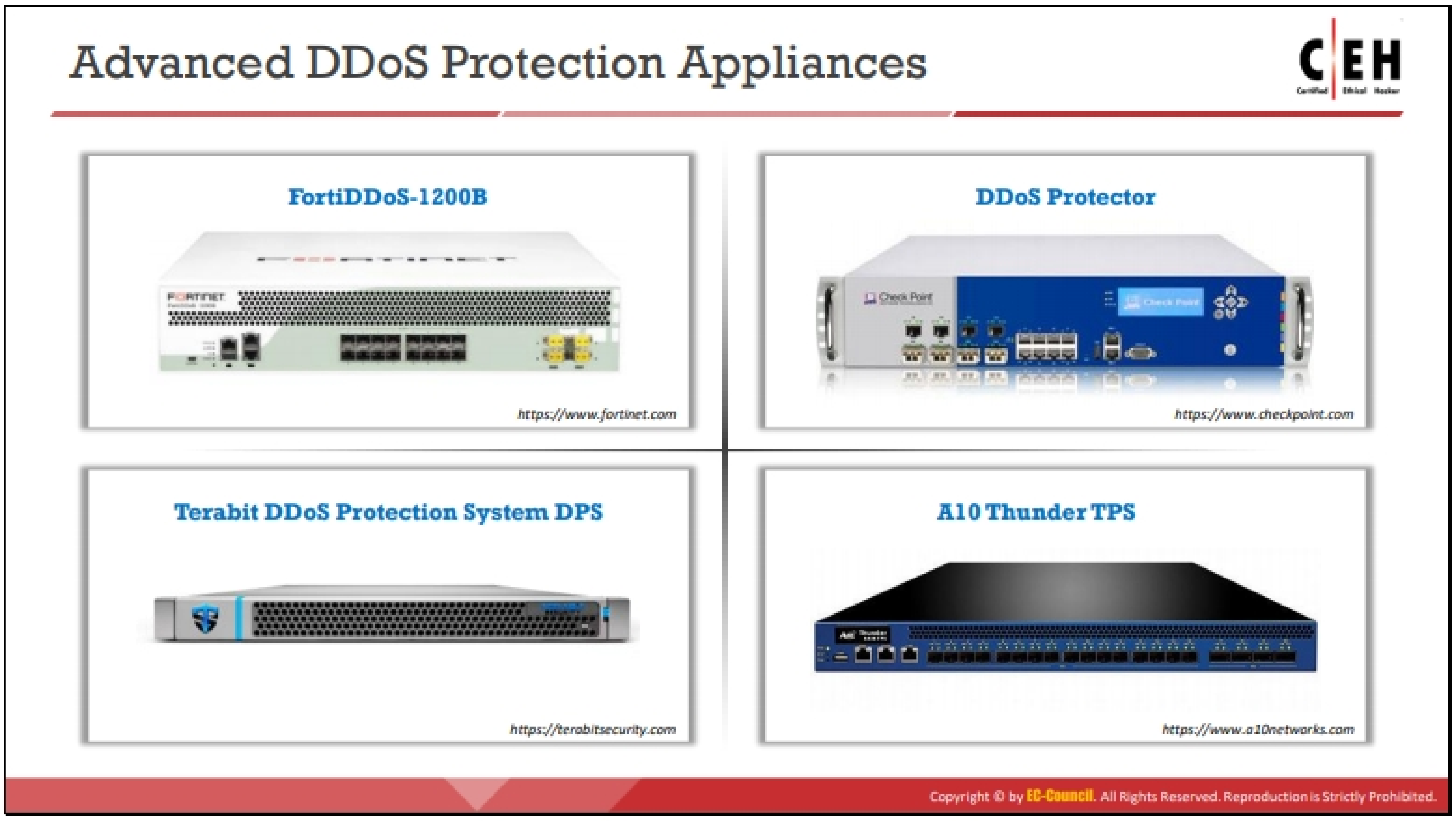


Module Flow

- DoS/DDoS Concepts
- DoS/DDoS Attack Techniques
- Botnets
- DDoS Case Study
- DoS/DDoS Attack Tools
- Counter-measures
- **DoS/DDoS Protection Tools**

DoS/DDoS Protection Tools

This section discusses various hardware and software DoS/DDoS protection tools such as FortiDDoS, DDoS Protector, Imperva Incapsula, and Anti DDoS Guardian that are effective in safeguarding networks from DoS/DDoS attacks.



Advanced DDoS Protection Appliances

FortiDDoS-1200B

https://www.fortinet.com

DDoS Protector

https://www.checkpoint.com

Terabit DDoS Protection System DPS

https://terabitsecurity.com

A10 Thunder TPS

https://www.a10networks.com

Advanced DDoS Protection Appliances

The following are examples for appliances that provide advanced protection against DDoS attacks.

- **FortiDDoS-1200B**

  Source: *https://www.fortinet.com*

  FortiDDoS provides comprehensive protection against DDoS attacks. It helps protect Internet infrastructure from threats and service disruptions by surgically removing network and application layer DDoS attacks while letting legitimate traffic flow without being impacted.

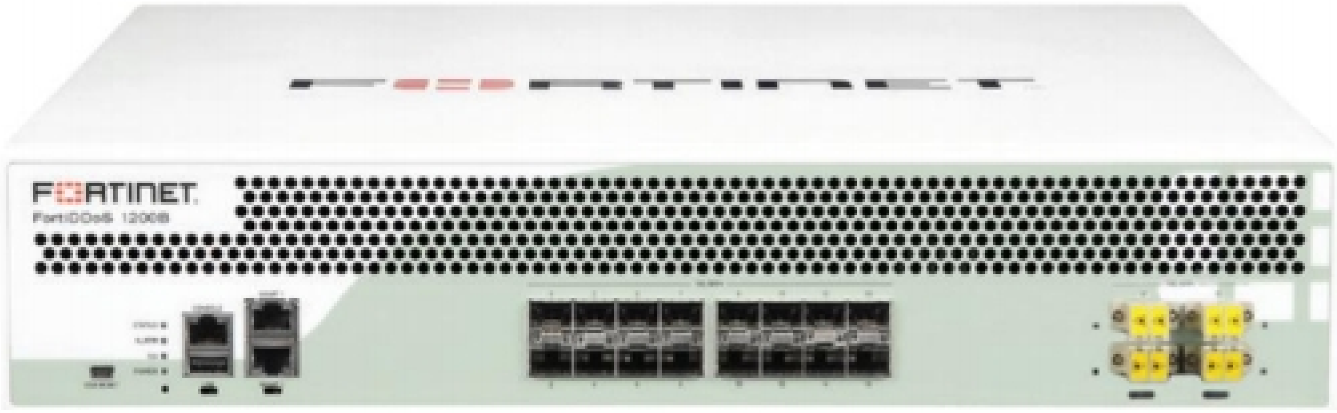


Figure 10.35: FortiDDoS-1200B

- **DDoS Protector**

  Source: *https://www.checkpoint.com*

  Check Point DDoS Protector blocks DDoS attacks with multi-layered protection. Its advantages are listed as follows:

  - Blocks a wide range of attacks with customized multi-layered protection

    - Behavioral protection base-lining multiple elements and blocking abnormal traffic
    - Automatically generated and predefined signatures
    - Use of advanced challenge/response techniques
  - Fast response time to protect against attacks within seconds
    - Automatically defends against network flood and application layer attacks
    - Customized protection optimized to meet the security needs of a specific network environment
    - Quickly filters traffic before it reaches the firewall to protect networks and servers as well as block exploits
  - Flexible deployment options to protect any business
  - Integrated with Check Point Security Management

Figure 10.36: DDoS Protector

- **Terabit DDoS Protection System**

  Source: *https://terabitsecurity.com*

  Terabit DDoS Protection System (DPS) is a solution for the detection and subsequent treatment of DDoS attacks. Terabit DPS helps ensure the maximum availability of a network and eliminates any disruptions caused by DoS/DDoS attacks. It can be used for large networks of bandwidth up to 1 Tbps. It can also provide protection for bandwidth up to 6.4 Tbps.

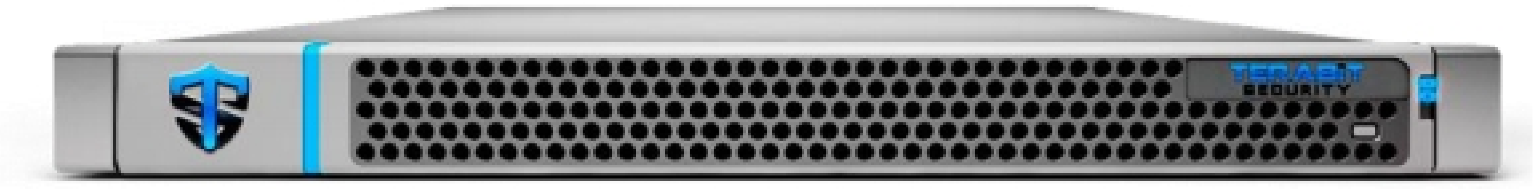


Figure 10.37: Terabit DPS appliance

- **A10 Thunder TPS**

  Source: *https://www.a10networks.com*

  A10 Thunder Threat Protection System (TPS) ensures reliable access to key network services by detecting and blocking external threats such as DDoS and other cyber-attacks before they escalate into costly service outages. Its features are listed as follows:

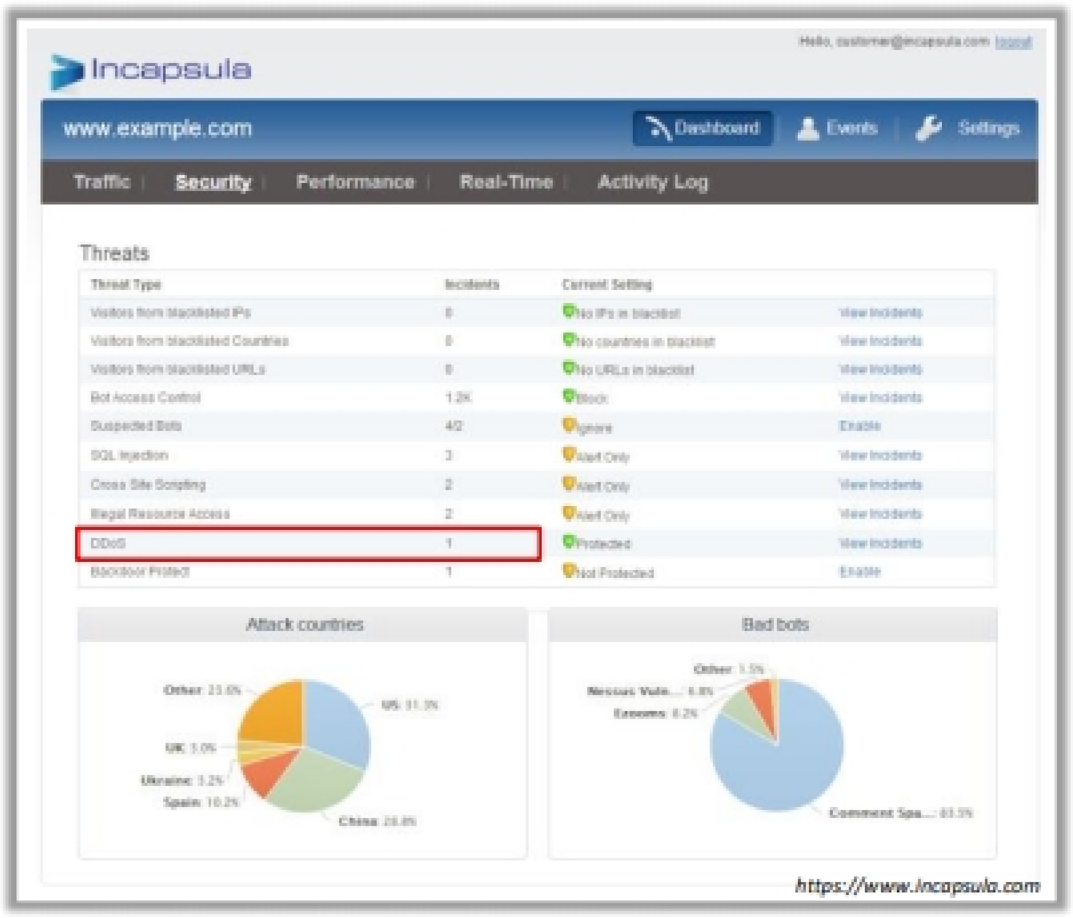
  - Custom protection with immediate blocking

- Proactive DDoS detection and mitigation
- Combined on-premises and cloud-based DDoS protection
- Built-in SSL inspection to block encrypted traffic
- Inbound reputation-based DDoS protection
- Inbound and outbound advanced threat protection

A10 Thunder

Figure 10.38: A10 Thunder TPS

DoS/DDoS Protection Tools


Incapsula
Hello, customer@incapsula.com logout
www.example.com
Dashboard
Events
Settings
Traffic
Security
Performance
Real-Time
Activity Log
Attack countries
Other: 23.6%
US: 31.3%
UK: 3.0%
Ukraine: 3.2%
Spain: 10.2%
China: 28.8%
Bad bots
Other: 1.5%
Nessus Vuln...: 6.8%
Ezooms: 8.2%
Comment Spa...: 83.5%

Threats

Threat Type	Incidents	Current Setting	
Visitors from blacklisted IPs	0	No IPs in blacklist	View incidents
Visitors from blacklisted Countries	0	No countries in blacklist	View incidents
Visitors from blacklisted URLs	0	No URLs in blacklist	View incidents
Bot Access Control	1.2K	Block	View incidents
Suspected Bots	4/2	Ignore	Enable
SQL Injection	3	Alert Only	View incidents
Cross Site Scripting	2	Alert Only	View incidents
Illegal Resource Access	2	Alert Only	View incidents
DDoS	1	Protected	View incidents
Backdoor Protect	1	Not Protected	Enable

https://www.incapsula.com

Imperva Incapsula DDoS Protection

Imperva Incapsula DDoS protection quickly mitigates attacks of any size without affecting **legitimate traffic** or **increasing latency**

DoS/DDoS Protection Tools

- Anti DDoS Guardian (*http://www.beethink.com*)
- DOSarrest's DDoS protection service (*https://www.dosarrest.com*)
- DDoS-GUARD (*https://ddos-guard.net*)
- Cloudflare (*https://www.cloudflare.com*)
- F5 (*https://f5.com*)

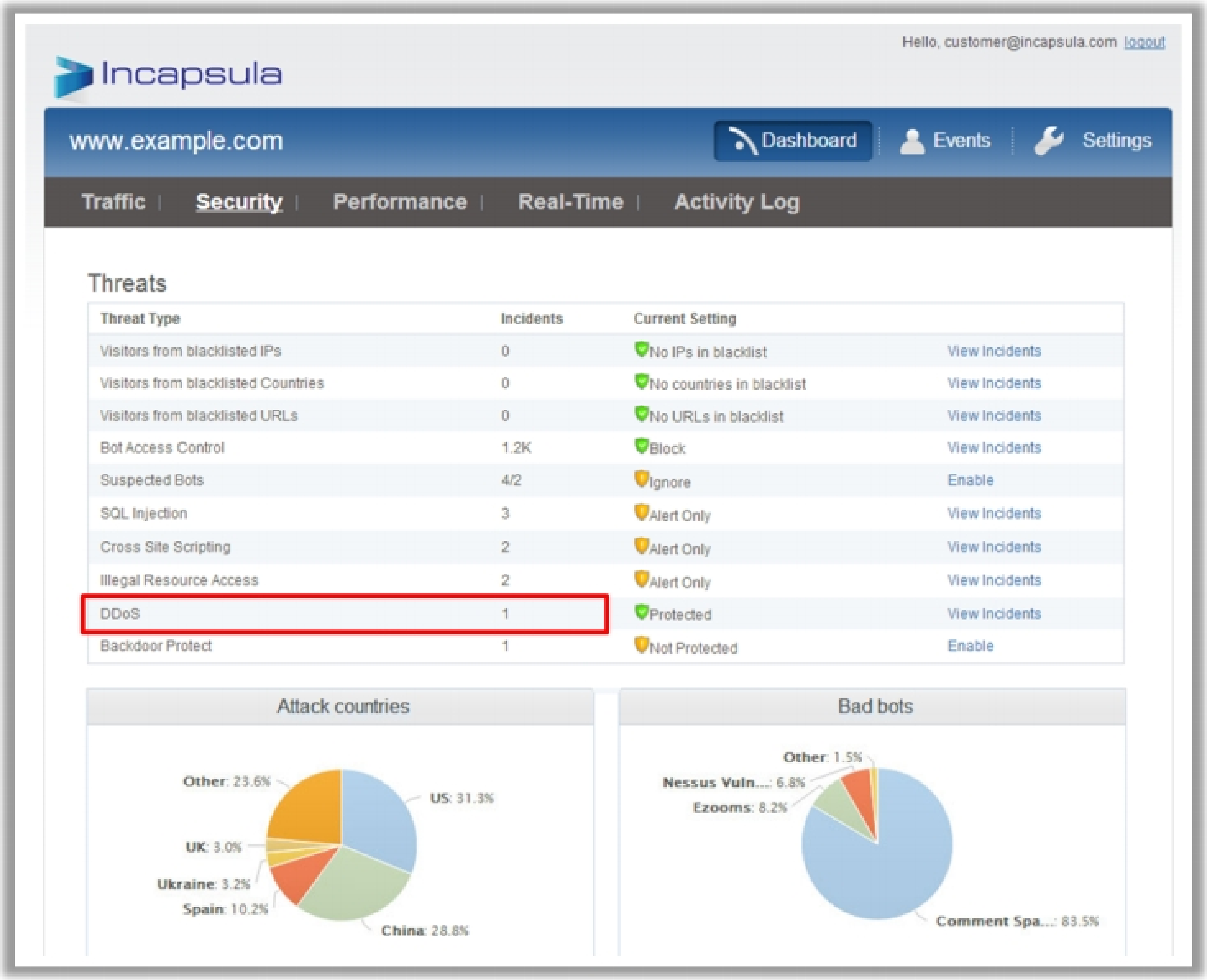
DoS/DDoS Protection Tools

- **Imperva Incapsula DDoS Protection**

  Source: *https://www.incapsula.com*

  Imperva Incapsula DDoS protection quickly mitigates any size attack without disrupting legitimate traffic or increasing latency. It is designed to provide multiple DDoS protection options and supports unicast and anycast technologies to power a many-to-many defense methodology. It automatically detects and mitigates attacks exploiting application and server vulnerabilities, hit-and-run events, and large botnets.

  Incapsula proxies all web requests to block DDoS attacks from being relayed to client origin servers. Incapsula detects and mitigates any type of attack, including TCP SYN + ACK, TCP FIN, TCP RESET, TCP ACK, TCP ACK + PSH, TCP fragmentation, UDP, Slowloris, spoofing, ICMP, IGMP, HTTP flood, brute force, connection flood, DNS flood, NXDomain, mixed SYN + UDP or ICMP + UDP flood, ping of death, reflected ICMP & UDP, and Smurf.


Hello, customer@incapsula.com logout
Incapsula
www.example.com
Dashboard
Events
Settings
Traffic
Security
Performance
Real-Time
Activity Log
Attack countries
Other: 23.6%
US: 31.3%
UK: 3.0%
Ukraine: 3.2%
Spain: 10.2%
China: 28.8%
Bad bots
Other: 1.5%
Nessus Vuln...: 6.8%
Ezooms: 8.2%
Comment Spa...: 83.5%

Threats

Threat Type	Incidents	Current Setting	
Visitors from blacklisted IPs	0	No IPs in blacklist	View Incidents
Visitors from blacklisted Countries	0	No countries in blacklist	View Incidents
Visitors from blacklisted URLs	0	No URLs in blacklist	View Incidents
Bot Access Control	1.2K	Block	View Incidents
Suspected Bots	4/2	Ignore	Enable
SQL Injection	3	Alert Only	View Incidents
Cross Site Scripting	2	Alert Only	View Incidents
Illegal Resource Access	2	Alert Only	View Incidents
DDoS	1	Protected	View Incidents
Backdoor Protect	1	Not Protected	Enable

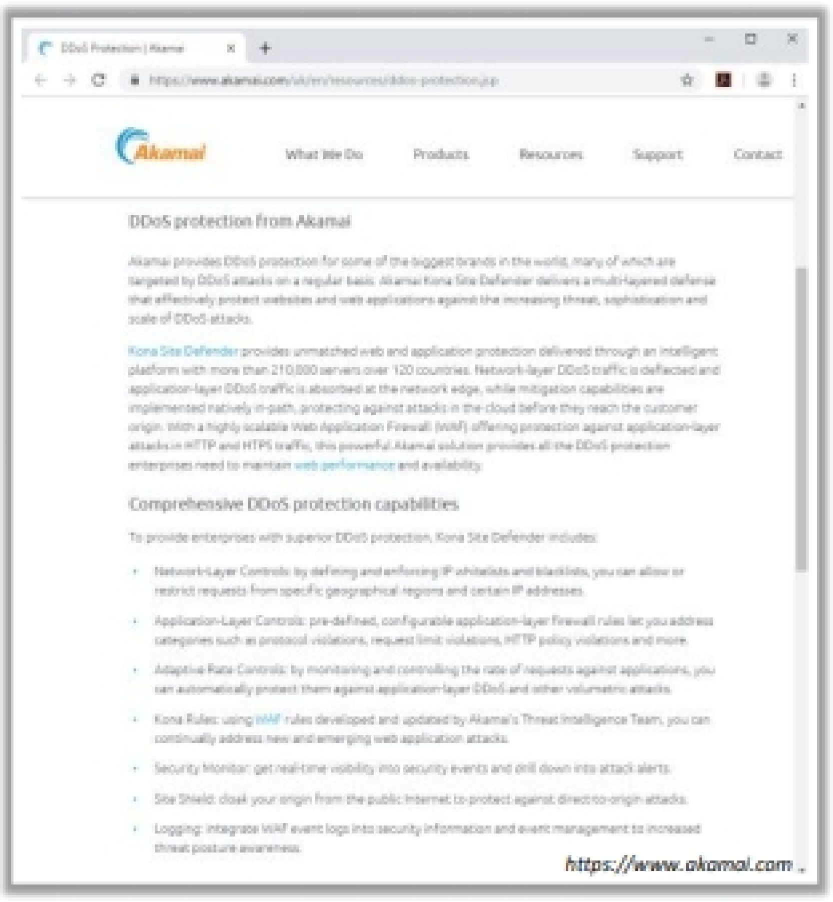
Figure 10.39: Screenshot of Incapsula DDoS protection tool

The following are examples for additional DDoS protection tools:

- Anti DDoS Guardian (*http://www.beethink.com*)
- DOSarrest's DDoS protection service (*https://www.dosarrest.com*)
- DDoS-GUARD (*https://ddos-guard.net*)
- Cloudflare (https://www.cloudflare.com)
- F5 (*https://f5.com*)

DoS/DDoS Protection Services

Akamai DDoS Protection



DDoS protection from Akamai

Akamai provides DDoS protection for some of the biggest brands in the world, many of which are targeted by DDoS attacks on a regular basis. Akamai Kona Site Defender delivers a multi-layered defense that effectively protect websites and web applications against the increasing threat, sophistication and scale of DDoS attacks.

Kona Site Defender provides unmatched web and application protection delivered through an intelligent platform with more than 210,000 servers over 120 countries. Network-layer DDoS traffic is deflected and application-layer DDoS traffic is absorbed at the network edge, while mitigation capabilities are implemented natively in-path, protecting against attacks in the cloud before they reach the customer origin. With a highly scalable Web Application Firewall (WAF) offering protection against application-layer attacks in HTTP and HTTPS traffic, this powerful Akamai solution provides all the DDoS protection enterprises need to maintain web performance and availability.

Comprehensive DDoS protection capabilities

To provide enterprises with superior DDoS protection, Kona Site Defender includes:

- Network-Layer Controls: by defining and enforcing IP whitelists and blacklists, you can allow or restrict requests from specific geographical regions and certain IP addresses.
- Application-Layer Controls: pre-defined, configurable application-layer firewall rules let you address categories such as protocol violations, request limit violations, HTTP policy violations and more.
- Adaptive Rate Controls: by monitoring and controlling the rate of requests against applications, you can automatically protect them against application-layer DDoS and other volumetric attacks.
- Kona Rules: using WAF rules developed and updated by Akamai's Threat Intelligence Team, you can continually address new and emerging web application attacks.
- Security Monitor: get real-time visibility into security events and drill down into attack alerts.
- Site Shield: cloak your origin from the public Internet to protect against direct-to-origin attacks.
- Logging: integrate WAF event logs into security information and event management to increased threat posture awareness.

https://www.akamai.com

Kaspersky DDoS Protection Tool
https://www.kaspersky.com

Stormwall PRO
https://stormwall.pro

Corero Network Security
https://www.corero.com

Nexusguard
https://www.nexusguard.com

BlockDoS
https://www.blockdos.net

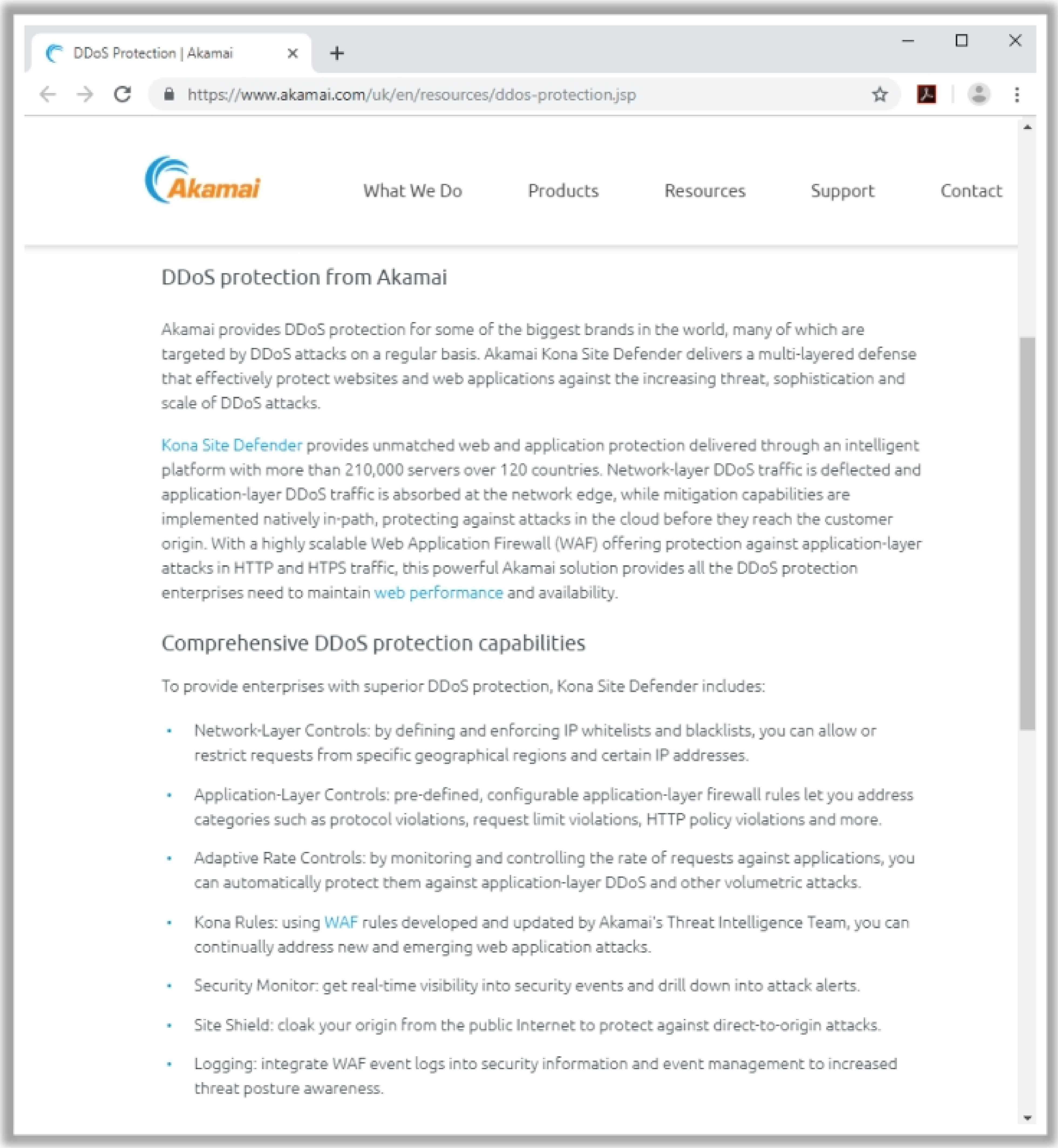
DoS/DDoS Protection Services

- **Akamai DDoS Protection**

  Source: *https://www.akamai.com*

  Akamai provides DDoS protection for enterprises regularly targeted by DDoS attacks. Akamai Kona Site Defender delivers multi-layered defense that effectively protects websites and web applications against the increasing threat, sophistication, and scale of DDoS attacks.

  Kona Site Defender provides unmatched web and application protection, which is delivered through an intelligent platform with more than 210,000 servers over 120 countries. Network-layer DDoS traffic is deflected and application layer DDoS traffic is absorbed in the network edge, while mitigation capabilities are implemented natively in-path, protecting against attacks in the cloud before they reach the client origin. Kona Site Defender has a highly scalable web application firewall (WAF) offering protection against application layer attacks in HTTP and HTTPS traffic, providing a complete DDoS protection solution for enterprises to maintain web performance and availability.



DDoS protection from Akamai

Akamai provides DDoS protection for some of the biggest brands in the world, many of which are targeted by DDoS attacks on a regular basis. Akamai Kona Site Defender delivers a multi-layered defense that effectively protect websites and web applications against the increasing threat, sophistication and scale of DDoS attacks.

Kona Site Defender provides unmatched web and application protection delivered through an intelligent platform with more than 210,000 servers over 120 countries. Network-layer DDoS traffic is deflected and application-layer DDoS traffic is absorbed at the network edge, while mitigation capabilities are implemented natively in-path, protecting against attacks in the cloud before they reach the customer origin. With a highly scalable Web Application Firewall (WAF) offering protection against application-layer attacks in HTTP and HTPS traffic, this powerful Akamai solution provides all the DDoS protection enterprises need to maintain web performance and availability.

Comprehensive DDoS protection capabilities

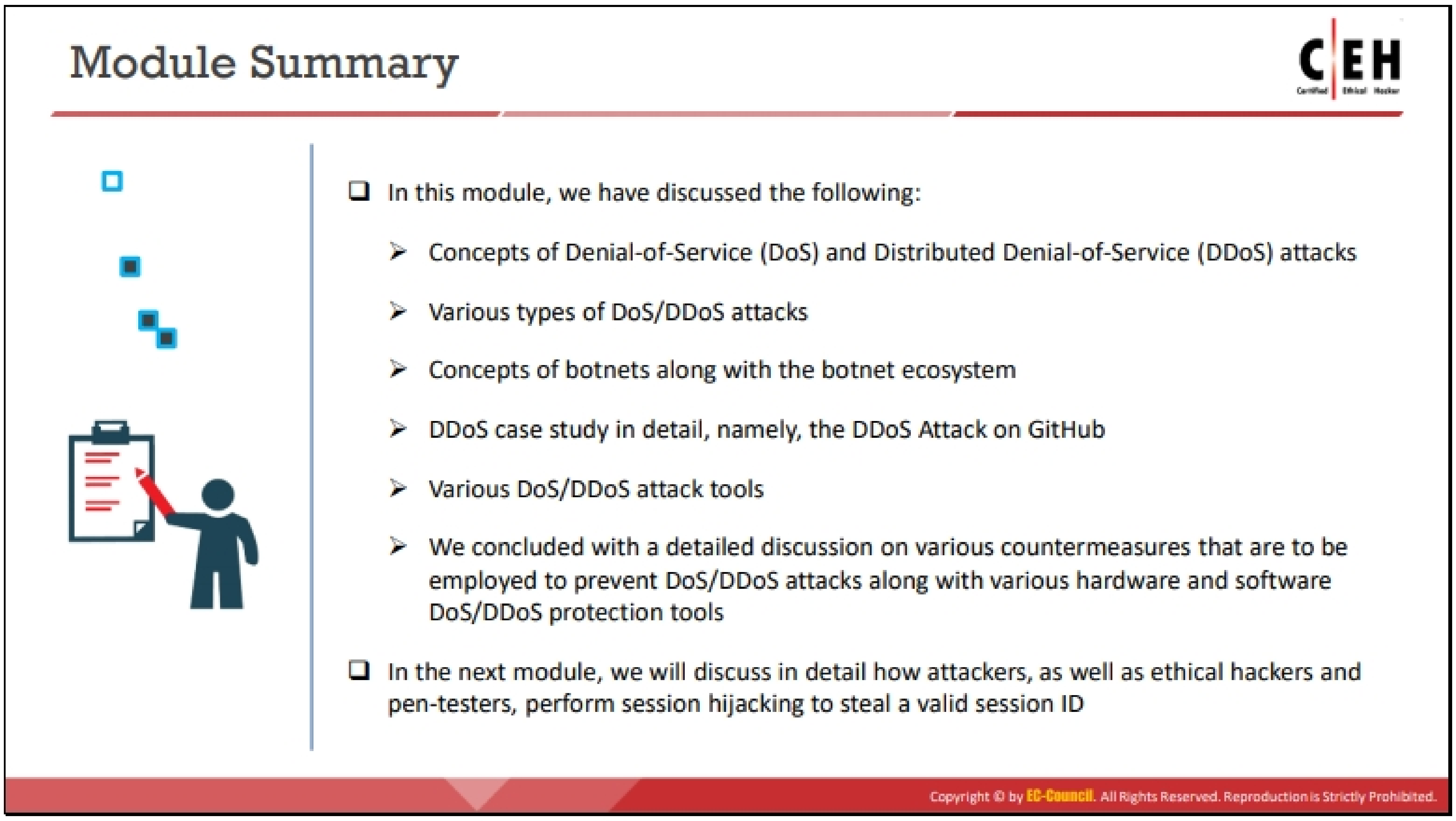
To provide enterprises with superior DDoS protection, Kona Site Defender includes:

- Network-Layer Controls: by defining and enforcing IP whitelists and blacklists, you can allow or restrict requests from specific geographical regions and certain IP addresses.
- Application-Layer Controls: pre-defined, configurable application-layer firewall rules let you address categories such as protocol violations, request limit violations, HTTP policy violations and more.
- Adaptive Rate Controls: by monitoring and controlling the rate of requests against applications, you can automatically protect them against application-layer DDoS and other volumetric attacks.
- Kona Rules: using WAF rules developed and updated by Akamai's Threat Intelligence Team, you can continually address new and emerging web application attacks.
- Security Monitor: get real-time visibility into security events and drill down into attack alerts.
- Site Shield: cloak your origin from the public Internet to protect against direct-to-origin attacks.
- Logging: integrate WAF event logs into security information and event management to increased threat posture awareness.

Figure 10.40: Akamai DDoS protection service

The following are examples for additional DDoS protection services:

- Kaspersky DDoS Protection Tool (*https://www.kaspersky.com*)
- Stormwall PRO (*https://stormwall.pro*)
- Corero Network Security (*https://www.corero.com*)
- Nexusguard (*https://www.nexusguard.com*)
- BlockDoS (*https://www.blockdos.net*)



Module Summary

- In this module, we have discussed the following:
    - Concepts of Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks
    - Various types of DoS/DDoS attacks
    - Concepts of botnets along with the botnet ecosystem
    - DDoS case study in detail, namely, the DDoS Attack on GitHub
    - Various DoS/DDoS attack tools
    - We concluded with a detailed discussion on various countermeasures that are to be employed to prevent DoS/DDoS attacks along with various hardware and software DoS/DDoS protection tools
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform session hijacking to steal a valid session ID

Module Summary

In this module, we discussed concepts related to denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks. We also discussed various types of DoS/DDoS attacks. Additionally, this module discussed concepts related to botnets along with the botnet ecosystem. Further, a detailed case study of a DDoS attack on GitHub was presented. Moreover, this module illustrated various DoS/DDoS attack tools and concluded with a detailed discussion on various countermeasures to prevent DoS/DDoS attacks, along with various hardware and software DoS/DDoS protection tools.

In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform session hijacking to steal a valid session ID.